L Number	Hits	Search Text	DB	Time stamp
1	109	"mutual exclusion" and (thread$1 or process$2 or (multiple near2 thread$1)) same owner$1	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:08
2	7	("mutual exclusion" and (thread$1 or process$2 or (multiple near2 thread$1)) same owner$1) and 707/$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:04
3	25	"mutual exclusion" and ((thread$1 or process$2 or (multiple near2 thread$1)) same owner$1 same (acquir$4 or inquir$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:12
4	44	("mutual exclusion" or mutexes) and ((thread$1 or process$2 or (multiple near2 thread$1)) same owner$1 same (acquir$4 or inquir$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:20
5	5	(("mutual exclusion" or mutexes) and ((thread$1 or process$2 or (multiple near2 thread$1)) same owner$1 same (acquir$4 or inquir$4))) and 707/$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:21
6	44	("mutual exclusion" or mutexes) and ((thread$1 or job$1 or process$2 or (multiple near2 thread$1)) same owner$1 same (acquir$4 or inquir$4))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:20
7	5	(("mutual exclusion" or mutexes) and ((thread$1 or job$1 or process$2 or (multiple near2 thread$1)) same owner$1 same (acquir$4 or inquir$4))) and 707/$.ccls.	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB	2004/07/20 14:21

Lock reservation: Java locks can mostly do without atomic operations

Full text Pdf (246 KB)

Source **Conference on Object Oriented Programming Systems Languages and Applications** archive
Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications table of contents
Seattle, Washington, USA
SESSION: Optimizations table of contents
Pages: 130 - 141
Year of Publication: 2002
ISBN:1-58113-471-1
Also published in ...

Authors Kiyokuni Kawachiya IBM Research, Tokyo Research Laboratory, Shimotsuruma, Yamato, Kanagawa, Japan
Akira Koseki IBM Research, Tokyo Research Laboratory, Shimotsuruma, Yamato, Kanagawa, Japan
Tamiya Onodera IBM Research, Tokyo Research Laboratory, Shimotsuruma, Yamato, Kanagawa, Japan

Sponsors ACM: Association for Computing Machinery
SIGPLAN: ACM Special Interest Group on Programming Languages

Publisher ACM Press New York, NY, USA

DOI Bookmark: Use this link to bookmark this Article: http://doi.acm.org/10.1145/582419.582433
What is a DOI?

↑ ABSTRACT

Because of the built-in support for multi-threaded programming, Java programs perform many lock operations. Although the overhead has been significantly reduced in the recent virtual machines, One or more atomic operations are required for acquiring and releasing an object's lock even in the fastest cases.This paper presents a novel algorithm called *lock reservation*. It exploits *thread locality* of Java locks, which claims that the locking sequence of a Java lock contains a very long repetition of a specific thread. The algorithm allows locks to be reserved for threads. When a thread attempts to acquire a lock, it can do without any atomic operation if the lock is reserved for the thread. Otherwise, it cancels the reservation and falls back to a conventional locking algorithm.We have evaluated an implementation of lock reservation in IBM's production virtual machine and compiler. The results show that it achieved performance improvements up to 53% in real Java programs.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full text article. ACM has opted to expose the complete List rather than only correct and linked references.

1 Sarita V. Adve , Kourosh Gharachorloo, Shared Memory Consistency Models: A Tutorial,

h c g c cf b c c

Computer, v.29 n.12, p.66-76, December 1996

2 Ole Agesen , David Detlefs , Alex Garthwaite , Ross Knippel , Y. S. Ramakrishna , Derek White, An efficient meta-lock for implementing ubiquitous synchronization, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.207-222, November 01-05, 1999, Denver, Colorado, United States

3 Jonathan Aldrich , Craig Chambers , Emin Gün Sirer , Susan J. Eggers, Static Analyses for Eliminating Unnecessary Synchronization from Java Programs, Proceedings of the 6th International Symposium on Static Analysis, p.19-38, September 22-24, 1999

4 E. Armstrong. HotSpot: A New Breed of Virtual Machine.

5 David Francis Bacon , Susan L. Graham, Fast and effective optimization of statically typed object-oriented languages, 1997

6 David F. Bacon , Ravi Konuru , Chet Murthy , Mauricio Serrano, Thin locks: featherweight synchronization for Java, Proceedings of the ACM SIGPLAN 1998 conference on Programming language design and implementation, p.258-268, June 17-19, 1998, Montreal, Quebec, Canada

7 D. F. Bacon and S. Fink. Personal Communication.

8 Brian N. Bershad , David D. Redell , John R. Ellis, Fast mutual exclusion for uniprocessors, Proceedings of the fifth international conference on Architectural support for programming languages and operating systems, p.223-233, October 12-15, 1992, Boston, Massachusetts, United States

9 Bruno Blanchet, Escape analysis for object-oriented languages: application to Java, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.20-34, November 01-05, 1999, Denver, Colorado, United States

10 Jeff Bogda , Urs Hölzle, Removing unnecessary synchronization in Java, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.35-46, November 01-05, 1999, Denver, Colorado, United States

11 Peter A. Buhr , Michel Fortier , Michael H. Coffin, Monitor classification, ACM Computing Surveys (CSUR), v.27 n.1, p.63-107, March 1995

12 Jong-Deok Choi , Manish Gupta , Mauricio Serrano , Vugranam C. Sreedhar , Sam Midkiff, Escape analysis for Java, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.1-19, November 01-05, 1999, Denver, Colorado, United States

13 D. Dice. Implementing Fast Java Monitors with Relaxed-Locks. Proceedings of USENIX JVM '01, 79--90, 2001.

14 E. W. Dijkstra, Solution of a problem in concurrent programming control, Communications of the ACM, v.8 n.9, p.569, Sept. 1965

15 R. Eckstein, D. Collier-Brown, and P. Kelly. Using Samba. O'Reilly, 1999. http://www.oreilly.com/catalog/samba/chapter/book/ch05_05.html.

16 E. M. Gagnon and L. J. Hendren. SableVM: A Research Framework for the Efficient Execution of Java Bytecode Proceedings of USENIX JVM '01, 27--39, 2001.

17 James Gosling , Bill Joy , Guy L. Steele, The Java Language Specification, Addison-Wesley

h c g c cf b c c

Longman Publishing Co., Inc., Boston, MA, 1996

18 C. A. R. Hoare, Monitors: an operating system structuring concept, Communications of the ACM, v.17 n.10, p.549-557, Oct. 1974

19 Richard L. Hudson , J. Elliot Moss , Sreenivas Subramoney , Weldon Washburn, Cycles to recycle: garbage collection to the IA-64, Proceedings of the second international symposium on Memory management, p.101-110, October 15-16, 2000, Minneapolis, Minnesota, United States

20 IBM developerWorks Java Technology Zone. http://www.ibm.com/developerworks/java/.

21 Intel Corporation. IA-32 Intel Architecture Software Developer's Manual Vol. 1--3. http://developer.intel.com/design/Pentium4/manuals/.

22 Intel Corporation. Intel Itanium Architecture Software Developer's Manual Vol. 1--3. http://developer.intel.com/design/itanium/manuals/.

23 Kazuaki Ishizaki , Motohiro Kawahito , Toshiaki Yasue , Mikio Takeuchi , Takeshi Ogasawara , Toshio Suganuma , Tamiya Onodera , Hideaki Komatsu , Toshio Nakatani, Design, implementation, and evaluation of optimizations in a just-in-time compiler, Proceedings of the ACM 1999 conference on Java Grande, p.119-128, June 12-14, 1999, San Francisco, California, United States

24 Java Community Process. JSR 133: Java Memory Model and Thread Specification Revision. http://jcp.org/jsr/detail/133.jsp.

25 T. Johnson and K. Harathi. Interruptible Critical Sections. Technical Report TR94007, University of Florida, 1994.

26 Kaffe.org. Developing Kaffe. http://www.kaffe.org/develop.html.

27 M. Kawahito. Personal Communication.

28 A. Krall and M. Probst. Monitors and Exceptions: How to Implement Java Efficiently. Proceedings of ACM Workshop on Java for High-Performance Network Computing, 15--24, 1998.

29 H. T. Kung , John T. Robinson, On optimistic methods for concurrency control, ACM Transactions on Database Systems (TODS), v.6 n.2, p.213-226, June 1981

30 Leslie Lamport, A fast mutual exclusion algorithm, ACM Transactions on Computer Systems (TOCS), v.5 n.1, p.1-11, Feb. 1987

31 P. Leach and D. Perry. CIFS: A Common Internet File System. http://www.microsoft.com/mind/1196/cifs.asp|, 1996.

32 G. Muller, B. Moura, F. Bellard, and C. Consel. Harissa: A Flexible and Efficient Java Environment Mixing Bytecode and Compiled Code. Proceedings of the 3rd USENIX Conference on Object Oriented Technologies and Systems (COOTS '97), 1--20, 1997.

33 T. Onodera. A Simple and Space-Efficient Monitor Optimization for Java. IBM Research Report RT0259, IBM, 1998.

34 Tamiya Onodera , Kiyokuni Kawachiya, A study of locking objects with bimodal fields, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.223-237, November 01-05, 1999, Denver, Colorado, United States

35 Young Gil Park , Benjamin Goldberg, Escape analysis on lists, Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation, p.116-127, June 15-19, 1992, San Francisco, California, United States

36 William Pugh, Fixing the Java memory model, Proceedings of the ACM 1999 conference on Java Grande, p.89-98, June 12-14, 1999, San Francisco, California, United States

37 Ravi Rajwar , James R. Goodman, Speculative lock elision: enabling highly concurrent multithreaded execution, Proceedings of the 34th annual ACM/IEEE international symposium on Microarchitecture, December 01-05, 2001, Austin, Texas

38 Erik Ruf, Effective synchronization removal for Java, Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation, p.208-218, June 18-21, 2000, Vancouver, British Columbia, Canada

39 Standard Performance Evaluation Corporation. SPEC JBB2000. http://www.spec.org/osg/jbb2000/.

40 Standard Performance Evaluation Corporation. SPEC JVM98 Benchmarks. http://www.spec.org/osg/jvm98/.

41 T. Suganuma, T. Ogasawara, M. Takeuchi, T. Yasue, M. Kawahito, K. Ishizaki, H. Komatsu, and T. Nakatani. Overview of the IBM Java Just-in-Time Compiler. IBM Systems Journal, 39(1), 175--193, 2000.

42 Alexander Thomasian, Concurrency control: methods, performance, and analysis, ACM Computing Surveys (CSUR), v.30 n.1, p.70-119, March 1998

43 Volano LLC. Volano Benchmarks. http://www.volano.com/benchmarks.html.

44 John Whaley , Martin Rinard, Compositional pointer and escape analysis for Java programs, Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, p.187-206, November 01-05, 1999, Denver, Colorado, United States

45 John Whaley, Partial method compilation using dynamic profile information, Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications, p.166-179, October 14-18, 2001, Tampa Bay, FL, USA

46 Byung-Sun Yang , Junpyo Lee , Jinpyo Park , Soo-Mook Moon , Kemal Ebcioğlu , Erik Altman, Lightweight monitor for Java VM, ACM SIGARCH Computer Architecture News, v.27 n.1, p.35-38, March 1999

47 F. Yellin and T. Lindholm. Java Runtime Internals. Presentation in JavaOne '96,

↑ **CITINGS 2**

David F. Bacon , Ravi Konuru , Chet Murthy , Mauricio J. Serrano, Thin locks: featherweight Synchronization for Java, ACM SIGPLAN Notices, v.39 n.4, April 2004

Bruno Dufour , Karel Driesen , Laurie Hendren , Clark Verbrugge, Dynamic metrics for java, ACM SIGPLAN Notices, v.38 n.11, November 2003

↑ INDEX TERMS

Primary Classification:
D. Software
↳ **D.3** PROGRAMMING LANGUAGES
↳ **D.3.4** Processors
↳ **Subjects:** Optimization

General Terms:
Algorithms, Experimentation, Languages, Measurement, Performance

Keywords:
Java, atomic operation, lock, monitor, reservation, synchronization, thread locality

↑ Collaborative Colleagues:

Kiyokuni Kawachiya:	Osamu Gohda	Kazunori Ogata
	Tatsushi Inagaki	Masanobu Ogata
	Hiroshi Ishikawa	Tamiya Onodera
	Kazuaki Ishizaki	Toshio Suganuma
	Motohiro Kawahito	Mikio Takeuchi
	Hideaki Komatsu	Hideyuki Tokuda
	Akira Koseki	Toshiaki Yasue
	Toshio Nakatani	
	Nobuhiko Nishio	
	Takeshi Ogasawara	
Akira Koseki:	Yoshiaki Fukazawa	Tamiya Onodera
	Osamu Gohda	Toshio Suganuma
	Tatsushi Inagaki	Mikio Takeuchi
	Kazuaki Ishizaki	Toshiaki Yasue
	Kiyokuni Kawachiya	
	Motohiro Kawahito	
	Hideaki Komatsu	
	Toshio Nakatani	
	Takeshi Ogasawara	
	Kazunori Ogata	
Tamiya Onodera:	Osamu Gohda	Kazunori Ogata
	Tatsushi Inagaki	Toshio Suganuma
	Kazuaki Ishizaki	Mikio Takeuchi
	Kiyokuni Kawachiya	Toshiaki Yasue
	Motohiro Kawahito	
	Satoru Kawai	
	Hideaki Komatsu	
	Akira Koseki	
	Toshio Nakatani	
	Takeshi Ogasawara	

↑ Peer to Peer - Readers of this Article have also read:

- Web application security assessment by fault injection and behavior monitoring
  Proceedings of the twelfth international conference on World Wide Web
  Yao-Wen Huang , Shih-Kun Huang , Tsung-Po Lin , Chung-Hung Tsai

- Data structures for quadtree approximation and compression
  Communications of the ACM 28, 9
  Hanan Samet

- A hierarchical single-key-lock access control using the Chinese remainder theorem
  Proceedings of the 1992 ACM/SIGAPP Symposium on Applied computing
  Kim S. Lee , Huizhu Lu , D. D. Fisher

- 3D representations for software visualization
  Proceedings of the 2003 ACM symposium on Software visualization
  Andrian Marcus , Louis Feng , Jonathan I. Maletic

- Probabilistic surfaces: point based primitives to show surface uncertainty
  Proceedings of the conference on Visualization '02
  Gevorg Grigoryan , Penny Rheingans

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
  Volume 37 , Issue 11 (November 2002)

h c g c cf b c c

STIC EIC 2100 127611
Search Request Form

Today's Date: 07/20/04

What date would you like to use to limit the search?
Priority Date: 01/11/2000 Other:

Name TO, BAO QUOC
AU 2172 Examiner # 78889
Room # 4D42 Phone 305 9469
Serial # 09/480,340

Format for Search Results (Circle One):
PAPER DISK EMAIL

Where have you searched so far?
USP DWPI EPO JPO ACM IBM TDB
IEEE INSPEC SPI Other _____

Is this a "Fast & Focused" Search Request? (Circle One) YES NO
A "Fast & Focused" Search is completed in 2-3 hours (maximum). The search must be on a very specific topic and meet certain criteria. The criteria are posted in EIC2100 and on the EIC2100 NPL Web Page at http://ptoweb/patents/stic/stic-tc2100.htm.

What is the topic, novelty, motivation, utility, or other specific details defining the desired focus of this search? Please include the concepts, synonyms, keywords, acronyms, definitions, strategies, and anything else that helps to describe the topic. Please attach a copy of the abstract, background, brief summary, pertinent claims and any citations of relevant art you have found.

user
shareable resource
Mutex
multiple process
integrity
lockout

STIC Searcher Carolyn Phone 305-9729
Date picked up 7-20 Date Completed 7-21-04

sira Search and Information Resources Administration

```
File 348:EUROPEAN PATENTS 1978-2004/Jul W02
        (c) 2004 European Patent Office
File 349:PCT Fulltext 1979-2002/UB=20040708,UT=20040701
        (c) 2004 WIPO/Univentio

Set     Items   Description
S1     197724   OWN OR OWNS OR OWNED OR OWNER? OR OWNING OR COOWN?
S2       7652   MULTIJOB? ? OR MULTIPROCESS?? ? OR MULTITHREAD? OR MULTITA-
             SK?
S3     638010   THREAD?? ? OR PROCESSES OR JOB? ? OR TASK?? ? OR TASKING? ?
              OR PROGRAMMES OR PROGRAMS OR ENTITY? OR ENTITIES OR ROUTINE?
             ?
S4      41798   (MANY OR MULTIPLE OR MULTI OR SEVERAL OR NUMEROUS OR PLURA-
             LIT? OR MULTIPLE OR MULTITUD? OR PLURIF? OR BETWEEN)(1W)S3
S5      44669   (VARIOUS OR VARIETY OR GROUP???? ? OR CLUSTER? ? OR NUMBER
             OR SET OR SETS OR CHAIN? ? OR SERIES)(1W)S3
S6      34124   (TWO OR SECOND OR SECONDARY OR PAIR OR 2ND OR BOTH OR DUAL
             OR THREE OR TRIO OR THIRD OR 3RD)(1W)S3
S7     199733   LOCK OR LOCKS OR LOCKED OR LOCKING OR GHOSTLOCK? OR METALO-
             CK? OR SLEEPLOCK? OR SPINLOCK?
S8       1052   S1(3N)(INQUIR? OR QUERY? OR QUERIE? ? OR INTERROGAT? OR SU-
             BQUER? OR INQUIR? OR ENQUIR? OR ASK??? ? OR QUESTION??? ? OR -
             SEEK??? ?)
S9      16124   S1(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING
              OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCES-
             SING)
S10       165   S1(3N)(REUS???? ? OR REACQUISITION? OR REACCESS?)
S11      8180   MUTUAL?(1W)(EXCLUS? OR EXCLUD?) OR MUTEX? OR MUT()(EX OR E-
             XES OR EXING OR EXED)
S12      1624   SEMAPHORE? OR EVENTSEMAPHORE?
S13       418   CRITICAL()SECTION? ? OR CRITICAL()CODE? ?()SECTION? ?
S14       409   KERNELOBJECT? OR EVENTOBJECT? OR (KERNEL OR EVENT)()OBJECT?
              ?
S15      4611   RESOURCE? ?(1W)(MANAGE? OR MANAGING? OR DISPENS?)
S16    127853   SYNCHRONIS??????? ? OR SYNCHRONIZ?
S17     10088   S16(2N)(MECHANISM? OR OBJECT? ? OR STATE? ? OR PRIMITIVE? ?
              OR SERVICE? ? OR METHOD? ? OR STATEMENT? ?)
S18     19229   S16(2N)(OPERATION? ? OR FUNCTION? ? OR PROCESS?? ? OR DEVI-
             CE? ? OR UNIT OR UNITS)
S19    577627   DATA OR RESOURCE OR RESOURCES
S20    200177   S19(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USI-
             NG OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACC-
             ESSING)
S21       177   S8:S10(25N)(S2 OR MULTIPROCESS? OR S4:S6 OR MULTI()PROCESS-
             ??? ?)
S22         9   S21(25N)(S11:S15 OR S17:S18)
S23        12   S21(25N)S11:S18
S24        30   S21(25N)S20
S25        12   S22:S23
S26        12   IDPAT (sorted in duplicate/non-duplicate order)
S27        12   IDPAT (primary/non-duplicate records only)
S28        29   S24 NOT S27
S29        29   IDPAT (sorted in duplicate/non-duplicate order)
S30        28   IDPAT (primary/non-duplicate records only)
```

? t27/5,k/1-5,7,9,11

27/5,K/1 (Item 1 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01106326
Control of multiple computer processes
Steuerung von mehreren Rechnerprozessen
Commande des processus d'ordinateur multiple
PATENT ASSIGNEE:
Sun Microsystems, Inc., (2616592), 4150 Network Circle, Santa Clara, California 95054, (US), (Applicant designated States: all)
INVENTOR:
Williams, Emrys John, 1063 Morse Avenue no. 3-205, Sunnyvale, California 94089, (US)
LEGAL REPRESENTATIVE:
Harris, Ian Richard et al (72231), D. Young & Co., 21 New Fetter Lane, London EC4A 1DA, (GB)
PATENT (CC, No, Kind, Date): EP 969369 A2 000105 (Basic)
EP 969369 A3 040107
APPLICATION (CC, No, Date): EP 99304864 990622;
PRIORITY (CC, No, Date): US 107972 980630
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/46

ABSTRACT EP 969369 A2
A program controlled apparatus includes one or more units for executing a multiple process. A mutex ordering mechanism controls the ordering of mutex ownership to provide deterministic execution of the processes. A mutex processor monitors mutex registers for determining mutex ownership. The mutex registers can be configured as sets of mutex request registers and mutex release registers. The apparatus may include a single processor configured to execute multiple processes concurrently, or multiple processing units, each configured to execute one or more processes. A monitor unit which can monitor equivalent operation of the processing sets can also include the mutex ordering mechanism.
ABSTRACT WORD COUNT: 102
NOTE:
Figure number on first page: 12

LEGAL STATUS (Type, Pub Date, Kind, Text):
Assignee: 030423 A2 Transfer of rights to new applicant: Sun Microsystems, Inc. (2616592) 4150 Network Circle Santa Clara, California 95054 US
Application: 20000105 A2 Published application without search report
Search Report: 040107 A3 Separate publication of the search report
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200001	671
SPEC A	(English)	200001	13522
Total word count - document A			14193
Total word count - document B			0
Total word count - documents A + B			14193

...SPECIFICATION to achieve deterministic execution therefor. This can enable fault tolerance to be built into many **multi - process** (**multi - threaded**) processing environments including, for example, networked

fault tolerant systems.

A **mutex** processor can be operable to monitor **mutex** registers for determining **mutex ownership** . By controlling the **access** to the **mutex** registers, that is the ownership thereof, a deterministic ordering of **mutex** processing can be achieved.

The **mutex** registers can be configured as sets of mutex request registers and mutex release registers.

The...

27/5,K/2 (Item 2 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

00779185

Method and apparatus for crash safe enforcement of mutually exclusive access to shared resources in a multitasking computer system

Verfahren und Gerat zur absturzsicheren Durchfuhrung von gegenseitigem exklusivem Zugang zu gemeinsamen Betriebsmitteln in einer Multitaskrechnerumgebung

Procede et dispositif pour l'execution en toute securite mutuellement exclusif aux ressources partagees dans un environnement d'ordinateur multitaches

PATENT ASSIGNEE:
AT&T Corp., (589370), 32 Avenue of the Americas, New York, NY 10013-2412, (US), (applicant designated states: DE;FR;GB;SE)

INVENTOR:
Bohannon, Philip L., 10 Hilsinger Drive,P.O.Box 394, Mt.Tabor,New Jersey 07878,, (US)
Lieuwen, Daniel Francis, 229 Livingston Avenue, New Providence, New Jersey 07974, (US)
Gava, Jacques, 917B East Front Street, Plainfield, New Jersey 07062, (US)
Sudarshar, Sundararajarao, 229 Livingston Avenue, Floor 2, New Providence, New Jersey 07974, (US)

LEGAL REPRESENTATIVE:
Watts, Christopher Malcolm Kelway, Dr. et al (37391), Lucent Technologies (UK) Ltd, 5 Mornington Road, Woodford Green Essex, IG8 0TU, (GB)

PATENT (CC, No, Kind, Date): EP 727742 A2 960821 (Basic)
APPLICATION (CC, No, Date): EP 96300823 960207;
PRIORITY (CC, No, Date): US 390179 950217
DESIGNATED STATES: DE; FR; GB; SE
INTERNATIONAL PATENT CLASS: G06F-009/46;

ABSTRACT EP 727742 A2

A fast crash safe method and apparatus for enforcing mutually exclusive access to shared resources in a computer system through the use of semaphores. The acquisition and release of the semaphores is implemented at the user process level. An overestimate and underestimate of semaphore ownership are maintained in memory by library provided semaphore acquisition and release code. A cleanup routine reconciles the overestimate and underestimate to determine the ownership status of the semaphores. (see image in original document)

ABSTRACT WORD COUNT: 93

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 960821 A2 Published application (Alwith Search Report ;A2without Search Report)
Withdrawal: 980107 A2 Date on which the European patent application was withdrawn: 971024

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB96	1685
SPEC A	(English)	EPAB96	7100
Total word count - document A			8785
Total word count - document B			0
Total word count - documents A + B			8785

...CLAIMS whether the semaphore record is available,
b) an owner identification for registering ownership of the **semaphore** record, and
c) cleanup indicia for indicating whether the **semaphore** record is being evaluated by a cleanup means;
a **plurality** of **processes** executing on said at least one processing unit. said processes requesting and acquiring exclusive access to said plurality of shared resources by requesting and **acquiring ownership** of said **semaphore** records; and
a plurality of **semaphore** access records stored in said memory unit, each of said **semaphore** access records associated with one of said **plurality** of **processes** , for identifying a **semaphore** record which the process associated with said **semaphore** access record wants to **acquire ownership** of;
wherein said cleanup means further comprises means for determining whether a **semaphore** record is owned by a process which has crashed by using said information in said...

...at least one processing unit for storing a plurality of data structures;
a plurality of **semaphore** records stored in said memory unit;
a **plurality** of **processes** executing on said at least one processing unit, said processes requesting and acquiring exclusive access to said data structures by requesting and **acquiring ownership** of said **semaphore** records;
means for maintaining an underestimate of **semaphore** ownership in said memory unit; and
means for maintaining an overestimate of **semaphore** ownership in said memory unit; and
cleanup means operating on said semaphore records for determining...

...including a memory unit, connected to said at least one processing unit;
a plurality of **semaphore** records stored in said memory unit;
a **plurality** of **processes** executing on said at least one processing unit, said processes requesting and acquiring exclusive access to said shared resources by requesting and **acquiring ownership** of said **semaphore** records;
means for maintaining an undersestimate of **semaphore** ownership in said memory unit;
means for maintaining an overestimate of **semaphore** ownership in said memory unit; and
cleanup means operating on said semaphore records for determining...

27/5,K/3 (Item 3 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00693431
Method and system for managing ownership of a released synchronization mechanism

System und Verfahren fur die Eigentumerverwaltung eines freigegebenen Synchronisationsmechanismus

Procede et systeme pour gerer l'allocation d'un mecanisme de synchronisation apres sa liberation

PATENT ASSIGNEE:
MICROSOFT CORPORATION, (749861), One Microsoft Way, Redmond, Washington 98052-6399, (US), (Proprietor designated states: all)
INVENTOR:
Toutonghi, Michael, 5413 Village Park Drive No.2838, Bellevue, Washington 98006, (US)
LEGAL REPRESENTATIVE:
Grunecker, Kinkeldey, Stockmair & Schwanhausser Anwaltssozietat (100721), Maximilianstrasse 58, 80538 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 661633 A1 950705 (Basic)
EP 661633 B1 011107
APPLICATION (CC, No, Date): EP 94120347 941221;
PRIORITY (CC, No, Date): US 176132 931223
DESIGNATED STATES: DE; FR; GB
INTERNATIONAL PATENT CLASS: G06F-009/46
CITED REFERENCES (EP B):
OPERATING SYSTEMS REVIEW (SIGOPS), vol.25, no.1, January 1991, NEW YORK, US pages 4 - 18 S. HALDAR AND D. K. SUBRAMANIAN: 'FAIRNESS IN PROCESSOR SCHEDULING IN TIME SHARING SYSTEMS'
H.M. DEITEL & M.S. KOGAN: 'The Design of OS/2' 1992 , ADDISON-WESLEY PUBLISHING COMPANY , READING, MASSACHUSETTS, USA * page 108, paragraph 2 - page 113, paragraph 2 * * page 115, paragraph 3 - page 117, last paragraph; figure 5.5 * * page 123, paragraph 3 - page 125, paragraph 2 * * page 132, paragraph 3 - page 135, last paragraph *
RESEARCH DISCLOSURE, no.316, August 1990, HAVANT, GB page 668 DISCLOSURE NUMBER 31691: 'Synchronization of Resource Access';

ABSTRACT EP 661633 A1

A method and system for managing ownership of a released synchronization mechanism is provided. In a preferred embodiment, a number of entities, such as threads, are attempting to acquire the synchronization mechanism when the synchronization mechanism becomes available. Each of the entities has a priority indicator that indicates the relative urgency of the attempt by the entity to acquire the synchronization mechanism. The method and system first identifies one of the entities attempting to acquire the synchronization mechanism that has the priority indicator that indicates that its attempt to acquire the synchronization mechanism is of the highest urgency. The method and system then determines whether any entity attempted to acquire the synchronization mechanism during a foregoing selected period of time. If an entity has attempted to acquire the synchronization mechanism during the selected period of time, then the method and system assigns ownership of the synchronization mechanism to the identified entity. If no entity has attempted to acquire the synchronization mechanism during the selected period of time, then the method and system defers the assignment of ownership of the synchronization mechanism to a later time. (see image in original document)
ABSTRACT WORD COUNT: 192
NOTE:
Figure number on first page: 4

LEGAL STATUS (Type, Pub Date, Kind, Text):

Grant:	011107	B1 Granted patent
Application:	950705	A1 Published application (A1with Search Report ;A2without Search Report)
Oppn None:	021030	B1 No opposition filed: 20020808
Examination:	960306	A1 Date of filing of request for examination:

960104
Examination: 990303 A1 Date of despatch of first examination report: 990114
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	EPAB95	2015
CLAIMS B	(English)	200145	1272
CLAIMS B	(German)	200145	1258
CLAIMS B	(French)	200145	1427
SPEC A	(English)	EPAB95	4994
SPEC B	(English)	200145	4856
Total word count - document A			7010
Total word count - document B			8813
Total word count - documents A + B			15823

...CLAIMS the synchronization mechanism and ending at the time that the formerly owning thread released the **synchronization mechanism** , assigning ownership of the **synchronization mechanism** to the identified one of the **plurality** of **threads** ; and

if no thread has attempted to acquire the **synchronization mechanism** during the period of time beginning at the time that the formerly **owning** thread **acquired** the **synchronization mechanism** and ending at the time that the formerly owning thread released the **synchronization mechanism** , deferring the assignment of ownership of the **synchronization mechanism** .

11. A method in a computer system for assigning the ownership of a synchronization mechanism...owning process acquired the synchronization mechanism and ending at the time that the formerly owning **process** released the **synchronization mechanism** assigning ownership of the **synchronization mechanism** to the identified one of the plurality of processes; and

if no process has attempted to acquire the **synchronization mechanism** during the period of time beginning at the time that the formerly owning **process** acquired the **synchronization mechanism** and ending at the time that the formerly owning **process** released the **synchronization mechanism** , deferring the assignment of ownership of the **synchronization mechanism** .

12. A method in a computer system for determining whether to reassign the ownership of...

27/5,K/4 (Item 4 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00545631
Arbitrating multiprocessor accesses to shared resources
Arbitrierung des Multiprozessorzugriffs zu gemeinsamen Mitteln
Arbitration d'acces multiprocesseur au ressources partagees
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392730), 2550 Garcia Avenue, Mountain View, CA 94043, (US), (Proprietor designated states: all)
INVENTOR:
Lockwood, James M., 3305 Jarvis Avenue, San Jose, California 95118, (US)
LEGAL REPRESENTATIVE:
Wombwell, Francis (46021), Potts, Kerr & Co. 15, Hamilton Square, Birkenhead Merseyside L41 6BR, (GB)
PATENT (CC, No, Kind, Date): EP 543560 A2 930526 (Basic)
EP 543560 A3 930728

EP 543560 B1 991222
APPLICATION (CC, No, Date): EP 92310300 921111;
PRIORITY (CC, No, Date): US 794573 911119
DESIGNATED STATES: DE; FR; GB; IT
INTERNATIONAL PATENT CLASS: G06F-009/46
CITED PATENTS (EP A): US 3676860 A
CITED PATENTS (EP B): US 3676860 A
CITED REFERENCES (EP A):
PATENT ABSTRACTS OF JAPAN vol. 12, no. 194 (P-713)7 June 1988;
CITED REFERENCES (EP B):
PATENT ABSTRACTS OF JAPAN vol. 12, no. 194 (P-713)7 June 1988 & JP-A-62 298 866 (FUJITSU) 25 December 1987;

ABSTRACT EP 543560 A2

In a multiprocessor computer system, an access request and an access grant register is provided for storing an access request and an access grant semaphore for each shared resource. The access request and grant semaphores having a number of access request and grant bits assigned to the processors. Additionally, circuits are provided for each access request register for setting/clearing individual access request bits, and simultaneous reading of all access request bits of the stored access request semaphore. Furthermore, coordinated request and grant masks that reflect the relative access priorities of the processors are provided for the processors to use in conjunction with the current settings of the access request and grant semaphores to determine whether a shared resource is granted to a lower priority processor and whether a shared resource is being requested by a higher priority processor. As a result, multiprocessor accesses to shared resources are arbitrated in a manner having a number of advantages over the prior art. (see image in original document)
ABSTRACT WORD COUNT: 167
NOTE:
Figure number on first page: 1

LEGAL STATUS (Type, Pub Date, Kind, Text):
Oppn None: 001206 B1 No opposition filed: 20000923
Application: 930526 A2 Published application (A1with Search Report ;A2without Search Report)
Search Report: 930728 A3 Separate publication of the European or International search report
Change: 930728 A2 International patent classification (change)
Examination: 940323 A2 Date of filing of request for examination: 940119
Examination: 970917 A2 Date of despatch of first examination report: 970730
Grant: 991222 B1 Granted patent
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	199951	2558
CLAIMS B	(German)	199951	2360
CLAIMS B	(French)	199951	2842
SPEC B	(English)	199951	4143
Total word count - document A			0
Total word count - document B			11903
Total word count - documents A + B			11903

...SPECIFICATION processor unsets its assigned access grant and request bits of the access grant and request **semaphores** .

As a result, the processors of a **multiprocessor** computer system may perform interprocessor communication via the shared resource. A

contending processor may determine its relative access priority and the current **owner** based on the **access** request and grant **semaphores** . The relative access priorities of the processors to a shared resource may also be changed...

27/5,K/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

00401210
Interprocessor communication
Ubertragung zwischen Prozessoren
Communication entre processeurs
PATENT ASSIGNEE:
International Business Machines Corporation, (200120), Old Orchard Road, Armonk, N.Y. 10504, (US), (applicant designated states: AT;BE;CH;DE;DK;ES;FR;GB;GR;IT;LI;LU;NL;SE)
INVENTOR:
Dinwiddie, John Monroe, Jr., 112 Pacer Circle, West Palm Beach, FL 33414, (US)
Grice, Lonnie Edward, 252 N.W. 44th Street, Boca Raton, FL 33431, (US)
Joyce, James Maurice, 1544 N.W. 9th Street, Boca Raton, FL 33486, (US)
Loffredo, John Mario, 2694 S.W. 14th Drive, Deerfield Beach, FL 33414, (US)
Sanderson, Kenneth Russell, 1132 Widgeon Road, West Palm Beach, FL 33414, (US)
Baker, Ernest Dysart, 12032 Deer Run, Raleigh, North Carolina 27614, (US)
LEGAL REPRESENTATIVE:
Bailey, Geoffrey Alan (27921), IBM United Kingdom Limited Intellectual Property Department Hursley Park, Winchester Hampshire SO21 2JN, (GB)
PATENT (CC, No, Kind, Date): EP 398697 A2 901122 (Basic)
EP 398697 A3 940202
EP 398697 B1 980902
APPLICATION (CC, No, Date): EP 90305312 900516;
PRIORITY (CC, No, Date): US 353115 890517
DESIGNATED STATES: AT; BE; CH; DE; DK; ES; FR; GB; GR; IT; LI; LU; NL; SE
INTERNATIONAL PATENT CLASS: G06F-015/16;
CITED PATENTS (EP A): EP 205949 A; US 3940743 A; US 4315310 A; US 4812975 A ; EP 192944 A; EP 95363 A; EP 26649 A

ABSTRACT EP 398697 A2
The functions of two virtual opening systems (e.g. S/370 VM, VSE or IX370 and S/88 OS) are merged into one physical system. Partner pairs of S/88 processors run the S/88 OS and handle the fault tolerant and single system image aspects of the system. One or more partner pairs of S/370 processors are coupled to corresponding S/88 processors directly and through the S/88 bus. Each S/370 processor is allocated form 1 to 16 megabytes of contiguous storage from the S/88 main storage. Each S/370 virtual operating system thinks its memory allocation starts at address 0, and it manages its memory through normal S/370 dynamic memory allocation and paging techniques. The S/370 is limit checked to prevent the S/370 from accessing S/88 memory space. The S/88 Operating System is the master over all system hardware and I/O devices. The S/88 processors across the S/370 address space in direct response to a S/88 application program so that the S/88 may move I/O data into the S/370 I/O buffers and process the S/370 I/O operations. The S/88 and S/370 peer processor pairs to execute their respective Operating Systems in a single system environment without significant rewriting of either operating system. Neither operating system is aware of the other operating system nor the other processor pairs. (see image in original document)

ABSTRACT WORD COUNT: 219

LEGAL STATUS (Type, Pub Date, Kind, Text):

Lapse:	010606	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980902, CH 19980902, LI 19980902, GR 19980902, SE 19981202,
Application:	901122	A2	Published application (A1with Search Report ;A2without Search Report)
Lapse:	031105	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980902, CH 19980902, LI 19980902, DK 19981202, ES 19980902, GR 19980902, SE 19981202,
Lapse:	020612	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980902, CH 19980902, LI 19980902, ES 19980902, GR 19980902, SE 19981202,
Examination:	910206	A2	Date of filing of request for examination: 901213
Search Report:	940202	A3	Separate publication of the European or International search report
Examination:	960904	A2	Date of despatch of first examination report: 960717
Grant:	980902	B1	Granted patent
Lapse:	990602	B1	Date of lapse of the European patent in a Contracting State: CH 980902, LI 980902
Lapse:	990602	B1	Date of lapse of the European patent in a Contracting State: CH 980902, LI 980902
Lapse:	990811	B1	Date of lapse of European Patent in a contracting state (Country, date): CH 19980902, LI 19980902, SE 19981202,
Oppn None:	990825	B1	No opposition filed: 19990603
Lapse:	990825	B1	Date of lapse of European Patent in a contracting state (Country, date): AT 19980902, CH 19980902, LI 19980902, SE 19981202,

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	9836	397
CLAIMS B	(German)	9836	352
CLAIMS B	(French)	9836	454
SPEC B	(English)	9836	71173
Total word count - document A			0
Total word count - document B			72376
Total word count - documents A + B			72376

27/5,K/7 (Item 7 from file: 349)

DIALOG(R)File 349:PCT Fulltext

01123180

METHOD AND APPARATUS FOR SERIALIZED MUTUAL EXCLUSION
PROCEDE ET APPAREIL D'EXCLUSION MUTUELLE NUMEROTEE

Patent Applicant/Assignee:
INTEL CORPORATION, 2200 Mission College Boulevard, Santa Clara, CA 95052, US, US (Residence), US (Nationality)
Inventor(s):
NARAD Charles, 5872 Arboretum Drive, Los Altos, CA 94024, US,

HUSTON Lawrence, 1477 Cherry Garden Lane, San Jose, CA 95125, US,
Legal Representative:
HARRIS Scott C (et al) (agent), Fish & Richardson P.C., Suite 500, 4350 La Jolla Village Drive, San Diego, CA 92122, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200444742 A2 20040527 (WO 0444742)
Application: WO 2003US34372 20031029 (PCT/WO US03034372)
Priority Application: US 2002293908 20021112
Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE EG ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NI NO NZ OM PG PH PL PT RO RU SC SD SE SG SK SL SY TJ TM TN TR TT TZ UA UG UZ VC VN YU ZA ZM ZW
(EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT RO SE SI SK TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: G06F-009/46
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 6792

English Abstract
A mechanism that associates a mutual exclusion lock with a shared data item and provides ownership of the mutual exclusion lock to multiple execution threads that execute code operating on the shared data item in a sequential

French Abstract
La presente invention concerne un mecanisme qui associe un verrouillage en exclusion mutuelle avec un element de donnees partagees et qui fournit la propriete de ce verrouillage en exclusion mutuelle a de multiples unites d'execution qui executent le code fonctionnant sur cet element de donnees partagees dans un ordre sequentiel.

Legal Status (Type, Date, Text)
Publication 20040527 A2 Without international search report and to be republished upon receipt of that report.

Fulltext Availability:
Detailed Description

Detailed Description
... of execution are sharing a location, multiple threads may attempt to obtain ownership of the **mutual exclusion** lock simultaneously.

Under such conditions, the order in which the threads obtain 15 the **mutual exclusion** lock ownership is not guaranteed.

When **multiple threads** of execution on one or more processors are sharing data, a **mutual exclusion** lock (" **mutex** ") is **used** to provide **ownership** of the shared data to only one agent at a time. The use of a **mutex** allows the thread that holds the **mutex** to make one or more modifications to the contents of a shared record, or a...

27/5,K/9 (Item 9 from file: 349)
DIALOG(R)File 349:PCT Fulltext

00781825
SYSTEM OF REUSABLE SOFTWARE PARTS AND METHODS OF USE
SYSTEME D'UNITES LOGICIELLES REUTILISABLES ET PROCEDES D'UTILISATION
Patent Applicant/Assignee:
Z-FORCE CORPORATION, 151 Kalmus Drive, Suite B-250, Costa Mesa, CA 92626, US, US (Residence), US (Nationality)
Inventor(s):
MILOUSHEV Vladimir I, 30802 Calle Barbosa, Laguna Nigel, CA 92677, US,
NICKOLOV Peter A, 158 Giotto, Irvine, CA 92614, US,
Legal Representative:
TACHNER Adam H (et al) (agent), Crosby, Heafey, Roach & May, Suite 2000, Two Embarcadero Center, San Francisco, CA 94111, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200114959 A2-A3 20010301 (WO 0114959)
Application: WO 2000US22694 20000816 (PCT/WO US0022694)
Priority Application: US 99149371 19990816; US 99149624 19990816
Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: G06F-009/46
International Patent Class: G06F-009/44
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 182432

English Abstract
A system of reusable software parts for designing and constructing software components, applications and entire systems by assembly. Parts for generating events, shaping, distributing and controlling flows of events and other interactions are included. Also included are parts for handling synchronization and desynchronization of events and other interactions between parts, as well as parts for handling properties, parameterizing and serializing components, applications and systems. In addition, innovative adapter parts for interfacing parts that are not designed to work together are included. The system includes a dynamic container for software parts which supports integration of dynamically changing sets of parts into statically defined structures of parts. Other reusable parts for achieving such integration are also included.

French Abstract
L'invention se rapporte a un systeme d'unites logicielles reutilisables permettant la conception et la fabrication de composants logiciels, d'applications et de systemes entiers par assemblage. L'invention se rapporte a des unites destinees a la generation d'evenements, a la mise en forme, la distribution et la regulation de flux d'evenements et autres interactions. L'invention se rapporte egalement a des unites permettant de gerer la synchronisation et la desynchronisation d'evenements et autres interactions entre des unites, ainsi que des unites permettant de

gerer des proprietes, de parametrer et de serialiser des composants, des applications et des systemes. L'invention se rapporte en outre a des unites adaptatrices novatrices destinees a servir d'interface entre des unites qui ne sont pas concues pour travailler ensemble. Ledit systeme comprend un contenant dynamique pour les unites logicielles, qui assure l'integration d'ensembles d'unites changeant dynamiquement au sein de structures d'unites definies de maniere statique. L'invention se rapporte en outre a des unites reutilisables destinees a la mise en oeuvre d'une telle integration.

Legal Status (Type, Date, Text)
Publication 20010301 A2 Without international search report and to be republished upon receipt of that report.
Examination 20010802 Request for preliminary examination prior to end of 19th month from priority date
Search Rpt 20021107 Late publication of international search report
Republication 20021107 A3 With international search report.

27/5,K/11 (Item 11 from file: 349)
DIALOG(R)File 349:PCT Fulltext

00389647 **Image available**
A SYSTEM AND METHOD FOR CONTROLLING ACCESS TO A PERIPHERAL DEVICE UTILIZING A SYNCHRONIZATION PRIMITIVE
SYSTEME ET PROCEDE DE CONTROLE DES ACCES A UN PERIPHERIQUE PAR UTILISATION D'UNE PRIMITIVE DE SYNCHRONISATION
Patent Applicant/Assignee:
ADVANCED MICRO DEVICES INC,
Inventor(s):
ROBERTS David G,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9730390 A1 19970821
Application: WO 96US16706 19961017 (PCT/WO US9616706)
Priority Application: US 96602199 19960216
Designated States: JP KR AT BE CH DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Main International Patent Class: G06F-009/46
International Patent Class: G06F-13:12
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 3740

English Abstract
Method and system aspects control access through a synchronization primitive to a peripheral device in a processing system. A processing system in accordance with the present invention includes at least one central processing unit (CPU), and at least one peripheral device coupled to the at least one CPU, where the at least one peripheral device includes a synchronization primitive for controlling acquisition by at least one thread of execution from the at least one CPU. In a system aspect for controlling access, the system includes a circuit for sending a first signal from a component of the processing system to a synchronization primitive within the peripheral device to determine a state of the synchronization primitive, and acquiring control of the peripheral device when the synchronization primitive is in a released state. In a method for gaining control of a peripheral device in a processing system, the method includes sending a first signal from a component of the processing system to a semaphore register within the

peripheral device to determine a bit pattern in the semaphore register, and acquiring control of the peripheral device when the semaphore register stores a first bit pattern.

French Abstract
La presente invention concerne un systeme et un procede de controle des acces a un peripherique d'un systeme informatique par utilisation d'une primitive de synchronisation. Le systeme informatique de l'invention est constitue d'au moins une unite centrale (UC) et d'au moins un peripherique couple a cette UC. Le peripherique integre une primitive de synchronisation servant a controler l'usage exclusif par un train d'execution lance depuis l'UC. Le systeme de controle des acces comporte un circuit assurant l'envoi d'un premier signal issu d'un composant du systeme informatique a destination d'une primitive de synchronisation integree au peripherique, lequel signal demande a connaitre l'etat de la primitive de synchronisation et, si la primitive de synchronisation est a l'etat libere, ce signal demande l'acces en usage exclusif au peripherique. Le procede de reservation exclusive du peripherique du systeme informatique consiste a envoyer un premier signal en provenance d'un composant du systeme informatique a destination d'un registre semaphore integre au peripherique, lequel signal demande a connaitre une configuration binaire dans le registre semaphore, et, si le registre semaphore contient une premiere configuration binaire, le signal demande la reservation du peripherique en usage exclusif.

Fulltext Availability:
Detailed Description

Detailed Description
... the object at any one time, and only the thread with ownership can release that **ownership** . Thus, **access** to a peripheral 1/0 device would sometimes be associated with a **semaphore** object to ensure that **two threads** do not try to control the 1/0 device at the same time and interfere with each other. Although **synchronization objects** do ensure a lack of interference, their implementation through the operating system slows down the...
?

30/5,K/1 **(Item 1 from file: 348)**
DIALOG(R)File 348:EUROPEAN PATENTS

01367871
Apparatus and method for collecting resources which became unnecessary
Anordnung und Verfahren zum Sammeln von nicht mehr benotigten Betriebsmitteln
Dispositif et methode pour collecter des ressources devenues non necessaires
PATENT ASSIGNEE:
MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD., (216883), 1006, Oaza-Kadoma, Kadoma-shi, Osaka 571-8501, (JP), (Applicant designated States: all)
INVENTOR:
Shiomi, Takakazu, 30-5 Tohda-cho, Hirakata-shi, Osaka-fu, 573-0045, (JP)
Hayama, Satoru, 2-12-506 Sumiyoshi-dai, Higashinada-ku, Kobe-shi, Hyogo-ken, 658-0062, (JP)
Hiramoto, Takeshi, 4-6-B201, Minamishouwa-machi, Kaita-cho, Aki-gun, Hiroshima-ken, 736-0065, (JP)
Kubooka, Yuko, 2-7-25-501, Hikari-machi, Higashi-ku, Hiroshima-shi, Hiroshima-ken, 732-0052, (JP)
Doi, Shigenori, 6-16-58 Kawauchi, Asaminami-ku, Hiroshima-shi, Hiroshima-ken, 731-0102, (JP)
LEGAL REPRESENTATIVE:
Crawford, Andrew Birkby et al (29761), A.A. Thornton & Co. 235 High Holborn, London WC1V 7LE, (GB)
PATENT (CC, No, Kind, Date): EP 1164485 A2 011219 (Basic)
APPLICATION (CC, No, Date): EP 2001304329 010515;
PRIORITY (CC, No, Date): JP 2000141492 000515; JP 2000141493 000515; JP 2000398746 001227
DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE; TR
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/50; G06F-009/46

ABSTRACT EP 1164485 A2
When requested by an application, a library unit provides a resource to the application, acquires an ID for identifying the application which has requested the resource, and stores a combination of a resource name of the provided resource and the application ID in a table. When an application is completed, the library unit receives an application ID of the completed application, detects a resource name corresponding to the received application ID in the table, and collects a resource specified by the resource name.
ABSTRACT WORD COUNT: 84
NOTE:
Figure number on first page: 6

LEGAL STATUS (Type, Pub Date, Kind, Text):
Application: 011219 A2 Published application without search report
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200151	4158
SPEC A	(English)	200151	14258
Total word count - document A			18416
Total word count - document B			0
Total word count - documents A + B			18416

...SPECIFICATION even when one application ends, the VM unit 33a can execute the next application without **reacquiring** the necessary

resources . Also, each class library unit may reserve resources necessary for its **own** operation **using** the system special thread. For example, memory for holding the table showing the correspondence **between** applications, **threads** , and listeners shown in FIG. 19 may be reserved using the system special thread.

The...

30/5,K/3 (Item 3 from file: 348)

DIALOG(R)File 348:EUROPEAN PATENTS

01051672

MEMORY OPTIMIZATION STATE

SPEICHEROPTIMIERUNGSZUSTAND

ETAT D'OPTIMISATION D'UNE MEMOIRE

PATENT ASSIGNEE:

UNISYS CORPORATION, (842794), Township Line and Union Meeting Roads P.O. Box 500, Blue Bell, PA 19424-0001, (US), (Proprietor designated states: all)

INVENTOR:

BAUMAN, Mitchell, A., 48 Indian Hills Drive, Circle Pines, MN 55014, (US)

NILSON, Arthur, J., 45 Guilford Circle, Phoenixville, PA 19460, (US)

MORRISSEY, Douglas, E., 3923 Short Hill Drive, Allentown, PA 18104, (US)

LEGAL REPRESENTATIVE:

Modiano, Guido, Dr.-Ing. et al (40786), Modiano, Josif, Pisanty & Staub, Baaderstrasse 3, 80469 Munchen, (DE)

PATENT (CC, No, Kind, Date): EP 1027655 A1 000816 (Basic)
EP 1027655 B1 040310
WO 1999023566 990514

APPLICATION (CC, No, Date): EP 98957585 981104; WO 98US23509 981104

PRIORITY (CC, No, Date): US 964626 971105

DESIGNATED STATES: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LI; LU; MC; NL; PT; SE

INTERNATIONAL PATENT CLASS: G06F-012/08

CITED PATENTS (EP B): EP 881579 A; WO 95/25306 A

CITED PATENTS (WO A): XP 78115

CITED REFERENCES (EP B):

"COMPACT GLOBAL TABLE FOR MANAGEMENT OF MULTIPLE CACHES" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 7, 1 December 1989, pages 322-324, XP000078115;

CITED REFERENCES (WO A):

"COMPACT GLOBAL TABLE FOR MANAGEMENT OF MULTIPLE CACHES" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 32, no. 7, 1 December 1989, pages 322-324, XP000078115;

NOTE:

No A-document published by EPO

LEGAL STATUS (Type, Pub Date, Kind, Text):

Application:	000816	A1	Published application with search report
Application:	990714	A1	International application (Art. 158(1))
Grant:	040310	B1	Granted patent
Change:	021023	A1	Title of invention (German) changed: 20020905
Examination:	000816	A1	Date of request for examination: 20000531
Change:	021030	A1	Title of invention (German) changed: 20020906

LANGUAGE (Publication,Procedural,Application): English; English; English

FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200411	290
CLAIMS B	(German)	200411	291
CLAIMS B	(French)	200411	338
SPEC B	(English)	200411	5052

Total word count - document A 0
Total word count - document B 5971
Total word count - documents A + B 5971

...SPECIFICATION defined by the appended claims is directed to a system and method for enabling a **multiprocessor** system employing a memory hierarchy to provide data **ownership** based upon previous **access** patterns of a **data** unit. The memory hierarchy comprises a main memory having a plurality of data units, a...
? t30/5,k/18,24,26

30/5,K/18 (Item 18 from file: 349)
DIALOG(R)File 349:PCT Fulltext

00841933 **Image available**
EFFICIENT IMPLEMENTATION OF SEVERAL INDEPENDENT STATE MACHINES IN THE SAME PROCESS
APPLICATION EFFICACE DE PLUSIEURS AUTOMATES FINIS INDEPENDANTS DANS LE MEME PROCESSUS
Patent Applicant/Assignee:
TELEFONAKTIEBOLAGET LM ERICSSON (publ), S-126 25 Stockholm, SE, SE (Residence), SE (Nationality)
Inventor(s):
BAKKE Knut, Tangen Alle 24, N-4817 His, NO,
EVENSEN Geir Olav, Havoddeien 29A, N-4823 Nedenes, NO,
MOHAGHEGHI Parastoo, Gjert Sorensens vei 30, N-4879 Grimstad, NO,
Legal Representative:
ERICSSON RADIO SYSTEMS AB (agent), Patent Unit Radio Access, S-164 80 Stockholm, SE,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200175601 A1 20011011 (WO 0175601)
Application: WO 2001SE598 20010320 (PCT/WO SE0100598)
Priority Application: NO 20001655 20000330
Designated States: AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CR CU CZ DE DK DM DZ EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR
(OA) BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Main International Patent Class: G06F-009/46
International Patent Class: H04L-029/00
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 2798

English Abstract
The invention relates to a process control in telecommunication systems, in particular GPRS. The invention is a system and a method for implementing several independent state machines in the same process. It describes how to organise the state machines in a tree, how they access their data and how to offer them some generic support, such as informing them of a common event.

French Abstract
L'invention concerne un controle de processus dans des systemes de

telecommunication, notamment le GPRS. L'invention est un systeme et un procede permettant de mettre en oeuvre plusieurs automates finis independants dans le meme processus. L'invention decrit la maniere d'organiser les automates finis en arbre, la maniere d'acceder a leurs donnees et la maniere de leur offrir quelque support generique, comme de les informer d'un evenement commun.

Legal Status (Type, Date, Text)
Publication 20011011 A1 With international search report.
Publication 20011011 A1 Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.
Examination 20011206 Request for preliminary examination prior to end of 19th month from priority date

Fulltext Availability:
Detailed Description

Detailed Description
... all modules on the same process is pos
sible by traversing the module tree
0 **Access** to the process **data** is organised; no module can
access **data** **owned** by another module
Number of **processes** started on a processor and, hence,
messages passing **between** the **processes** (which is costly for
a system with **many** **processes**) is reduced.

* A homogenous programming model is achieved
5 * It is easy to add new...

30/5,K/24 (Item 24 from file: 349)
DIALOG(R)File 349:PCT Fulltext

00747049 **Image available**
SYSTEM AND METHOD FOR ACCESSING PERSONAL INFORMATION
SYSTEME ET PROCEDE D'ACCES A DES DONNEES PERSONNELLES
Patent Applicant/Assignee:
RENSSELAER POLYTECHNIC INSTITUTE, 110 8th Street, Troy, NY 12180-3590, US , US (Residence), US (Nationality)
Inventor(s):
HSU Cheng, 7 Dalamar Court, Latham, NY 12110, US,
HUGHES Gregory N, 25 Old Farm Road, Chester, NJ 07930, US,
SZYMANSKI Boleslaw, P.O. Box 301, Newtonwille, NY 12128, US,
Legal Representative:
RUSSAVAGE Edward J (agent), Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210, US,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200060435 A2-A3 20001012 (WO 0060435)
Application: WO 2000US9265 20000407 (PCT/WO US0009265)
Priority Application: US 99128219 19990407
Designated States: CA JP
(EP) AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE
Main International Patent Class: G06F-001/00
International Patent Class: G06F-017/30
Publication Language: English
Filing Language: English
Fulltext Availability:
Detailed Description
Claims

Fulltext Word Count: 20722

English Abstract
A business model is provided wherein a consumer is in control of their own information. Consumer data may be controlled, for example, by a single personal system or by a plurality of systems acting in the interest of the person. A method is provided for differentiating between ownership and control of data, allowing more flexibility in distributing and controlling data. Further, a distributed system architecture is provided to distribute consumer data among a portable device and a plurality of general-purpose computer systems. The system may employ agents that perform transactions on behalf and in the best interests of the user while keeping personal data within the ownership and control of the person. A manager is provided to manage agents created by the system. In one aspect, a metadatabase is used to manage these agents. The portable device may also include a natural language user interface (NLUI or NLI) to enhance the ease of use. The portable device may use, for instance, a natural language query processor that minimized storage and processing required on the portable device.

French Abstract
L'invention concerne un modele de gestion dans lequel un client commande ses propres donnees. Les donnees du client peuvent etre commandees, par exemple, par un systeme personnel unique ou par plusieurs systemes agissant dans l'interet de la personne. Un procede de differenciation entre la propriete et la commande des donnees permet d'acquerir plus de souplesse dans la repartition et la commande des donnees. En outre, une architecture de systeme repartie permet de repartir les donnees de client entre un dispositif portatif et plusieurs systemes informatiques polyvalents. Le systeme peut faire intervenir des agents qui effectuent des transactions en faveur et dans l'interet de l'utilisateur tout en gardant les donnees personnelles sous le controle de la personne qui en reste proprietaire. Un gestionnaire est destine a gerer les agents crees par le systeme. Dans un aspect de l'invention, on utilise une meta-base de donnees pour gerer ces agents. Le dispositif portatif peut egalement comprendre une interface utilisateur en langage naturel (NLUI ou NLI) destinee a faciliter l'utilisation. Le dispositif portatif peut faire intervenir, par exemple, un processeur d'interrogation en langage naturel permettant de reduire la memoire et le traitement requis dans le dispositif portatif.

Legal Status (Type, Date, Text)
Publication 20001012 A2 Without international search report and to be republished upon receipt of that report.
Examination 20001228 Request for preliminary examination prior to end of 19th month from priority date
Search Rpt 20010412 Late publication of international search report
Republication 20010412 A3 With international search report.

Fulltext Availability:
Detailed Description
Claims

Detailed Description
... and control of data. The method comprises steps of operating a computer system having a **plurality** of **processes** , and wherein at least one of the processes executes as a user process; indicating, for **data accessed** by the at least one process, **ownership** of the **accessed data** ; and indicating, for the **data accessed** by the at least one process, control of the **accessed data** wherein indication of ownership and indication of control are independent. According to one embodiment of

...

Claim
... method for maintaining ownership and control of data, comprising:
operating a computer system having a **plurality** of **processes** , and wherein at least
one of the processes executes as a user process;
indicating, for **data** **accessed** by the at least one process, **ownership** of the
accessed **data** ; and
5 indicating, for the **data** **accessed** by the at least one process, control of the **accessed** **data** wherein indication of ownership and indication of control are independent.

21 The method according to...

30/5,K/26 (Item 26 from file: 349)
DIALOG(R)File 349:PCT Fulltext

00492214
MEMORY OPTIMIZATION STATE
ETAT D'OPTIMISATION D'UNE MEMOIRE
Patent Applicant/Assignee:
UNISYS CORPORATION,
Inventor(s):
BAUMAN Mitchell A,
NILSON Arthur J,
MORRISSEY Douglas E,
Patent and Priority Information (Country, Number, Date):
Patent: WO 9923566 A1 19990514
Application: WO 98US23509 19981104 (PCT/WO US9823509)
Priority Application: US 97964626 19971105
Designated States: AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW GH GM KE LS MW SD SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN GW ML MR NE SN TD TG
Main International Patent Class: G06F-012/08
Publication Language: English
Fulltext Availability:
Detailed Description
Claims
Fulltext Word Count: 5868

English Abstract
A system and method for enabling a **multiprocessor** system (100) employing a memory hierarchy to provide data **ownership** based upon previous **access** patterns of a **data** unit (230). The memory hierarchy comprises a main memory (210) having a plurality of data units (230), a plurality of caches (510) that operate independently of each other, and at least one coherent domain (530) interfaced to each cache (510). Each coherent domain (530) comprises at least two processors (540). The main memory (210) includes a directory (610) for maintaining information about each data unit (230). The main memory (210) maintains coherency of data among the plurality of caches (510) using the directory (610). The present invention allows a requesting agent (510, 540, 550), such as a processor (540) or cache (510, 550), to request a data unit (230) without

specifying the type of ownership, where ownership may be exclusive or shared. The directory (610) includes history information that defines the previous access pattern of the requested data unit (230), whereby main memory (210) checks, prior to forwarding the requested data unit (230) to the requesting agent (510, 540, 550), the history information to determine what type of ownership to associate with the requested data unit (230). The requested data unit (230) is then delivered to the requesting agent (510, 540, 550) with ownership rights specified by the history information.

French Abstract

La presente invention concerne un systeme et un procede permettant a un systeme multiprocesseur (100) d'utiliser une arborescence de memoire pour determiner la propriete de donnees sur la base des modeles d'acces anterieurs d'une unite de donnees (230). Cette arborescence de memoire comprend une memoire principale (210) possedant plusieurs unites de donnees (230), plusieurs antememoires (510) fonctionnant de maniere independante l'une de l'autre, et au moins un domaine coherent (530) en interface avec chacune des antememoires (510). Chaque domaine coherent (530) comprend au moins deux processeurs (540). La memoire principale (210) comprend un repertoire (610) pour contenir les informations concernant chacune des unites de donnees (230). La memoire centrale (210) maintient la coherence des donnees dans le groupe d'antememoires (510) utilisant le repertoire (610). Cette invention permet a un agent demandeur (510, 540, 550), tel qu'un processeur (540) ou une antememoire (510, 550), de demander une unite de donnees (230) sans specifier le type de propriete, la propriete pouvant etre exclusive ou partagee. Le repertoire (610) comprend les informations historiques qui definissent le modele d'acces anterieur de l'unite de donnees interrogee (230), la memoire centrale (210) s'occupant de verifier, avant d'acheminer l'unite de donnees demandee (230) a l'agent demandeur (510, 540, 550), les informations historiques de maniere a determiner le type de propriete a associer a l'unite de donnees demandee (230). L'unite de donnees demandee (230) est ensuite acheminee jusqu'a l'agent demandeur (510, 540, 550), les informations historiques ayant deja specifie les droits de propriete.

Fulltext Availability:
Detailed Description

English Abstract

A system and method for enabling a **multiprocessor** system (100) employing a memory hierarchy to provide data **ownership** based upon previous **access** patterns of a **data** unit (230). The memory hierarchy comprises a main memory (210) having a plurality of data...

Detailed Description

... Briefly stated, the present invention is directed to a system and method for enabling a **multiprocessor** system employing a memory hierarchy to provide data **ownership** based upon previous **access** patterns of a **data** unit. The memory hierarchy comprises a main memory having a plurality of data units, a...

?

File 347:JAPIO Nov 1976-2004/Mar(Updated 040708)
(c) 2004 JPO & JAPIO
File 350:Derwent WPIX 1963-2004/UD,UM &UP=200445
(c) 2004 Thomson Derwent

Set	Items	Description
S1	101030	OWN OR OWNS OR OWNED OR OWNER? OR OWNING OR COOWN?
S2	9746	MULTIJOB? ? OR MULTIPROCESS?? ? OR MULTITHREAD? OR MULTITASK?
S3	589880	THREAD?? ? OR PROCESSES OR JOB? ? OR TASK?? ? OR TASKING? ? OR PROGRAMMES OR PROGRAMS OR ENTITY? OR ENTITIES OR ROUTINE? ?
S4	15606	(MANY OR MULTIPLE OR MULTI OR SEVERAL OR NUMEROUS OR PLURALIT? OR MULTIPLE OR MULTITUD? OR PLURIF? OR BETWEEN)(1W)S3
S5	18929	(VARIOUS OR VARIETY OR GROUP???? ? OR CLUSTER? ? OR NUMBER OR SET OR SETS OR CHAIN? ? OR SERIES)(1W)S3
S6	14286	(TWO OR SECOND OR SECONDARY OR PAIR OR 2ND OR BOTH OR DUAL OR THREE OR TRIO OR THIRD OR 3RD)(1W)S3
S7	483278	LOCK OR LOCKS OR LOCKED OR LOCKING OR GHOSTLOCK? OR METALOCK? OR SLEEPLOCK? OR SPINLOCK?
S8	144	S1(3N)(INQUIR? OR QUERY? OR QUERIE? ? OR INTERROGAT? OR SUBQUER? OR INQUIR? OR ENQUIR? OR ASK??? ? OR QUESTION??? ? OR -SEEK??? ?)
S9	4277	S1(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCESSING)
S10	14	S1(3N)(REUS???? ? OR REACQUISITION? OR REACCESS?)
S11	813	MUTUAL?(1W)(EXCLUS? OR EXCLUD?) OR MUTEX? OR MUT()(EX OR EXES OR EXING OR EXED)
S12	524	SEMAPHORE? OR EVENTSEMAPHORE?
S13	323	CRITICAL()SECTION? ? OR CRITICAL()CODE? ?()SECTION? ?
S14	71	KERNELOBJECT? OR EVENTOBJECT? OR (KERNEL OR EVENT)()OBJECT? ?
S15	2629	RESOURCE? ?(1W)(MANAGE? OR MANAGING? OR DISPENS?)
S16	219172	SYNCHRONIS? OR SYNCHRONIZ?
S17	11164	S16(2N)(MECHANISM? OR OBJECT? ? OR STATE? ? OR PRIMITIVE? ? OR SERVICE? ? OR METHOD? ? OR STATEMENT? ?)
S18	22497	S16(2N)(OPERATION? ? OR FUNCTION? ? OR PROCESS?? ? OR DEVICE? ? OR UNIT OR UNITS)
S19	1568977	DATA OR RESOURCE OR RESOURCES
S20	209938	S19(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCESSING)
S21	100	S8:S10 AND (S2 OR MULTIPROCESS? OR S4:S6 OR MULTI()PROCESS-?)
S22	6	S21 AND S11:S18
S23	26	S21 AND S20
S24	29	S22:S23
S25	29	IDPAT (sorted in duplicate/non-duplicate order)
S26	29	IDPAT (primary/non-duplicate records only)

26/9/1 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

016088835 **Image available**
WPI Acc No: 2004-246710/200423
Related WPI Acc No: 2004-167969; 2004-167981
XRPX Acc No: N04-195662

Multi-target transactions providing method for computer system, involves committing transaction once ownership of each of targeted transactionable

locations has been acquired, using single-target synchronization primitive
Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: HERLIHY M; LUCHANGCO V M; MOIR M S
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20040034673	A1	20040219	US 2002396152	P	20020716	200423 B
			US 2003620747	A	20030716	

Priority Applications (No Type Date): US 2002396152 P 20020716; US 2003620747 A 20030716
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040034673	A1		13	G06F-012/00	Provisional application US 2002396152

Abstract (Basic): US 20040034673 A1

NOVELTY - The method involves attempting to **acquire ownership** of each of targeted transactionable locations for a particular transaction of **multi - threaded** computation. The transaction is committed once ownership of each of the targeted transactionable locations has been acquired, using a single-target **synchronization primitive** . The **primitive** ensures that the transaction continues to own each of the targeted transactionable locations.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) an implementation of non-blocking, multi-targets transactions that employs primitives to acquire, for a particular multi-target transaction

(2) a computer program product encoding

(3) a computer readable medium encoding a portion of a non-blocking, multi-target transaction implementation

(4) a multi-target transactions providing apparatus.

USE - Used for providing multi-target transactions in a computer system.

ADVANTAGE - The multi-target atomic transactions that target special transactionable locations in shared memory, are obstruction-free.

DESCRIPTION OF DRAWING(S) - The drawing shows a flow diagram that highlights major flows in execution of a multitarget compare and swap operation.

pp; 13 DwgNo 2/3

Title Terms: MULTI; TARGET; TRANSACTION; METHOD; COMPUTER; SYSTEM; TRANSACTION; LOCATE; ACQUIRE; SINGLE; TARGET; **SYNCHRONISATION** ; PRIMITIVE
Derwent Class: T01
International Patent Class (Main): G06F-012/00
File Segment: EPI
Manual Codes (EPI/S-X): T01-H03D; T01-S03

26/9/2 (Item 2 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015950001 **Image available**
WPI Acc No: 2004-107842/200411
XRPX Acc No: N04-085763
Computer system e.g. multiprocessing **computer system, changes** ownership **responsibility and** access **rights among coherency units at**

different timings based on received address and data packets
Patent Assignee: SUN MICROSYSTEMS INC (SUNM); CYPHER R E (CYPH-I)
Inventor: CYPHER R E
Number of Countries: 032 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20040003180	A1	20040101	US 2002184171	A	20020628	200411	B
EP 1380956	A2	20040114	EP 2003254042	A	20030625	200411	

Priority Applications (No Type Date): US 2002184171 A 20020628
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040003180	A1		44	G06F-012/00	
EP 1380956	A2	E		G06F-012/08	

Designated States (Regional): AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LT LU LV MC MK NL PT RO SE SI SK TR

Abstract (Basic): US 20040003180 A1
NOVELTY - An active device receives an address packet from a directory, indicating ownership responsibilities and performs transition of ownership responsibility for a specific coherency unit. The active device receives a data packet from another active device and performs transition of access rights for the coherency unit at a time different from the ownership transition time.
DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for coherency management method in computer system.
USE - Computer system e.g. **multiprocessing** computer system for use in various transactions through packet switched network and point-point network.
ADVANTAGE - Reduces processing latency during **data access** by utilizing the directory based coherency protocols effectively. Simplifies circuit configuration by minimizing the number of processors.
DESCRIPTION OF DRAWING(S) - The figure shows the schematic block diagram of computer system.
processing systems (142A,142B)
memory sub-systems (144A,144B)
input-output system (146)
pp; 44 DwgNo 1/20
Title Terms: COMPUTER; SYSTEM; **MULTIPROCESSOR** ; COMPUTER; SYSTEM; CHANGE; ACCESS; COHERE; UNIT; TIME; BASED; RECEIVE; ADDRESS; DATA; PACKET
Derwent Class: T01; W01
International Patent Class (Main): G06F-012/00; G06F-012/08
File Segment: EPI
Manual Codes (EPI/S-X): T01-H01C3; T01-M02; T01-N02B1; W01-A03B; W01-A05B; W01-A06E

26/9/3 (Item 3 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015633466 **Image available**
WPI Acc No: 2003-695648/200366
XRPX Acc No: N03-555433
Multitask **computing system operating method involves advising currently** owning **task to relinquish** use **of target** resource **item and is removed from in-use registry, when pending of takeaway request is determined**
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)

Inventor: LE C M; PENCE J W; RATLIFF J M
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 6604160	B1	20030805	US 2000671971	A	20000928	200366	B

Priority Applications (No Type Date): US 2000671971 A 20000928
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6604160	B1		17	G06F-013/38	

Abstract (Basic): US 6604160 B1

NOVELTY - A use-requesting task is entered into in-use registry (110) of target resource item, only if conflict resolution solution identified from priority matrix (111) allows the task to obtain target item, else takeaway request is submitted. When pending of takeaway request is determined, in response to arrival of recovery events, currently owning task is advised to relinquish **resource** item **usage** and is removed from in- use registry.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(1) computing system;

(2) data storage system;

(3) storage medium storing program for performing operations to arbitrate access among **multiple tasks** to shared computing resource items; and

(4) logic circuit for performing operations to arbitrate access among **multiple tasks** to shared computing resource items.

USE - For arbitrating **access** to computing **resources** such as magnetic tape cartridges or tape reels, cache and non-volatile memories, tape drive, optical disk drive, network **resources** , internet **access** , files, **data** sets, database records and server resources, in **multitask** computing environment.

ADVANTAGE - Prevents lengthy delays incurred, when a use- seeking task waits until the natural conclusion of a preceding task of **using** serially-accessible **data** storage media. Also permits the access-seeking tasks of higher priority to start **using** shared **resources** by takeaway operations, even if the task owning the resource fails.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of the data storage system.

applications (103-105)
storage manager (108)
in-use registry (110)
priority matrix (111)
controllers (112)
pp; 17 DwgNo 1/6

Title Terms: COMPUTATION; SYSTEM; OPERATE; METHOD; ADVICE; CURRENT; TASK; TARGET; RESOURCE; ITEM; REMOVE; REGISTER; PENDING; REQUEST; DETERMINE
Derwent Class: T01; T03; U21
International Patent Class (Main): G06F-013/38
File Segment: EPI
Manual Codes (EPI/S-X): T01-F02C2; T01-N02B1A; T01-S03; T03-N01; U21-C03B9

26/9/5 (Item 5 from file: 350)

014997638 **Image available**

WPI Acc No: 2003-058153/200305
XRPX Acc No: N03-045169
Cache data block requester prediction method in multiprocessor **system, involves informing data block requesting processor about processor currently owning data block**
Patent Assignee: LAI K (LAIK-I); PEIR J (PEIR-I); INTEL CORP (ITLC)
Inventor: LAI K; PEIR J
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20020144063	A1	20021003	US 2001823251	A	20010329	200305	B
US 6711662	B2	20040323	US 2001823251	A	20010329	200421	

Priority Applications (No Type Date): US 2001823251 A 20010329
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020144063	A1		9	G06F-012/00	
US 6711662	B2			G06F-012/00	

Abstract (Basic): US 20020144063 A1
NOVELTY - A cache data block requesting processor (121) is predicted. The requesting processor is informed about the processor currently **owning** the **data** block, **using ownership** history information table (23) stored in memory (16).
DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:
(1) Cache management system; and
(2) Computer program product storing cache data block requester prediction program.
USE - For predicting the processor requesting the cache data block, in **multiprocessor** system.
ADVANTAGE - By predicting more than one requester, the prediction accuracy can be improved. As the requester is informed about the current owner of the data block, the requester can directly request the **owner** , thus **data access** speed and throughput are increased.
DESCRIPTION OF DRAWING(S) - The figure shows the components of the cache for predicting data block requesters.
Memory (16)
Ownership history information table (23)
Processor (121)
pp; 9 DwgNo 2/3
Title Terms: CACHE; DATA; BLOCK; PREDICT; METHOD; **MULTIPROCESSOR** ; SYSTEM; INFORMATION; DATA; BLOCK; REQUEST; PROCESSOR; PROCESSOR; CURRENT; DATA; BLOCK
Derwent Class: T01
International Patent Class (Main): G06F-012/00
File Segment: EPI
Manual Codes (EPI/S-X): T01-H03A; T01-H08; T01-M02A1; T01-S03

26/9/7 (Item 7 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014708221 **Image available**
WPI Acc No: 2002-528925/200256
XRPX Acc No: N02-418863
Controlling access to data **segments by** acquiring **and transferring** ownership **of cache lines and** enacting storage locks
Patent Assignee: VARTTI K S (VART-I); UNISYS CORP (BURS)

Inventor: VARTTI K S
Number of Countries: 021 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200254250	A2	20020711	WO 2001US48743	A	20011212	200256 B
US 20020174305	A1	20021121	US 2000750637	A	20001228	200279
US 6625698	B2	20030923	US 2000750637	A	20001228	200364

Priority Applications (No Type Date): US 2000750637 A 20001228
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200254250	A2	E	36	G06F-012/08	
	Designated States (National): JP				
	Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR				
US 20020174305	A1			G06F-012/08	
US 6625698	B2			G06F-012/00	

Abstract (Basic): WO 200254250 A2

NOVELTY - Method consists in **acquiring ownership** status of the memory target data segments (TDSs) (cache lines), enacting a storage lock (SL) by prohibiting other requesting services from acting on the TDSs while the targeted (cache) memory owns the TDSs, issuing an SL release signal when exclusivity is no longer required, and releasing the SL to allow other devices to act on the TDSs. Segment ownership is based on temporary exclusivity requirements and ownership can be transferred before the SL is released.

DETAILED DESCRIPTION - There are INDEPENDENT CLAIMS for:

(1) A system for maintaining cache coherency

(2) A computer program for managing storage locks

(3) An apparatus for controlling **access** to **data** segments to maintain memory coherency

USE - Method is for controlling storage locks based on cache line ownership in **multiprocessor** data processing systems.

DESCRIPTION OF DRAWING(S) - The figure shows a computing system block diagram.

pp; 36 DwgNo 1/4

Title Terms: CONTROL; ACCESS; DATA; SEGMENT; ACQUIRE; TRANSFER; CACHE; LINE ; STORAGE; LOCK
Derwent Class: T01
International Patent Class (Main): G06F-012/00; G06F-012/08
International Patent Class (Additional): G06F-013/00
File Segment: EPI
Manual Codes (EPI/S-X): T01-H03A; T01-H05B1; T01-S03

26/9/8 (Item 8 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014621689 **Image available**
WPI Acc No: 2002-442393/200247
XRPX Acc No: N02-348389

Operation control method in multiprocessor **computer system, involves** acquiring ownership **of memory location by a microprocessor to perform read, write or modification operations without any interference**

Patent Assignee: INTEL CORP (ITLC)
Inventor: CARMEAN D M; KUMAR H; LINCE B E; UPTON M D; ZHANG Z
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week

US 6370625 B1 20020409 US 99474698 A 19991229 200247 B

Priority Applications (No Type Date): US 99474698 A 19991229
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 6370625 B1 10 G06F-013/18

Abstract (Basic): US 6370625 B1

NOVELTY - Ownership of a memory location is acquired by a microprocessor (12) to perform read, write or modification operations, when correct data is stored in that location. Other operations directed towards the data stored in that location are prevented, when operations are performed by the processor. The ownership is released after operation completion to allow the processor to perform operations in different memory location.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) Microprocessor;

(2) Computer system; and

(3) Machine readable medium with operation control program.

USE - For **synchronization** of processing of memory instructions by microprocessors (claimed) or **between** their **threads** in computer system (claimed).

ADVANTAGE - The microprocessor or its thread has ownership of the lock while it is being processed, thereby precluding other microprocessors or threads from **accessing** the **data** . Hence, successful read, write or modification operation is ensured without any interference from other microprocessors.

DESCRIPTION OF DRAWING(S) - The figure shows the block diagram of microprocessor.

Microprocessor (12)

pp; 10 DwgNo 2/4

Title Terms: OPERATE; CONTROL; METHOD; **MULTIPROCESSOR** ; COMPUTER; SYSTEM; ACQUIRE; MEMORY; LOCATE; MICROPROCESSOR; PERFORMANCE; READ; WRITING; MODIFIED; OPERATE; INTERFERENCE
Derwent Class: T01
International Patent Class (Main): G06F-013/18
File Segment: EPI
Manual Codes (EPI/S-X): T01-H01C2; T01-H03D; T01-H05B1; T01-M02; T01-S03

26/9/9 (Item 9 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014244941 **Image available**
WPI Acc No: 2002-065641/200209
Related WPI Acc No: 2002-033829
XRPX Acc No: N02-048755

Hybrid shared-nothing/shared disk database system for multiprocessing **computer system, allows direct** access **to** data **in** ownership **groups only for processes executed on servers and members of owner set**

Patent Assignee: ORACLE CORP (ORAC-N)
Inventor: PUTZOLU G
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
US 6321238 B1 20011120 US 98222577 A 19981228 200209 B

Priority Applications (No Type Date): US 98222577 A 19981228
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 6321238	B1		15	G06F-017/30	

Abstract (Basic): US 6321238 B1

NOVELTY - Several database servers execute on corresponding nodes, where each node has direct access to persistent storage devices. A portion of database is divided into ownership groups where each group is assigned with an owner **set** . The **processes** that are executed on servers which are members of the owner set, are only allowed to directly **access data** within the group.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are also included for the following:

(a) Database access management method;

(b) Computer readable medium for storing instructions for managing database access

USE - For use in **multiprocessing** computer system.

ADVANTAGE - Provides a single database system that is able to provide the performance advantages for both shared-nothing/shared disk database architectures. Provides better fault containment in case of software failures and frequent read access errors.

DESCRIPTION OF DRAWING(S) - The figure shows the flowchart for processing a data item in ownership groups supporting system.

pp; 15 DwgNo 3/5

Title Terms: HYBRID; SHARE; SHARE; DISC; DATABASE; SYSTEM; **MULTIPROCESSOR** ; COMPUTER; SYSTEM; ALLOW; DIRECT; ACCESS; DATA; GROUP; PROCESS; EXECUTE; SERVE; MEMBER; OWNER; SET

Derwent Class: T01

International Patent Class (Main): G06F-017/30

International Patent Class (Additional): G06F-012/00

File Segment: EPI

Manual Codes (EPI/S-X): T01-J05B4A; T01-J05B4M; T01-N02A3C; T01-N02B1; T01-S03

26/9/10 **(Item** 10 from file: 350)

DIALOG(R)File 350:Derwent WPIX

013783105 **Image available**

WPI Acc No: 2001-267316/200128

Related WPI Acc No: 1989-186482; 1989-186483; 1991-001580; 1991-001581; 1991-310785; 1991-348214; 1991-348215; 1992-034292; 1992-041979; 1993-102775; 1994-127941; 1999-432322; 2000-484542; 2000-507062; 2000-550699; 2000-648613; 2002-270962

XRPX Acc No: N01-191252

Memory management for digital data processing contains number of processing cells has data item identifier for all of data items in system address space which is unitary system address space

Patent Assignee: SUN MICROSYSTEMS INC (SUNM)

Inventor: BURKHARDT H; FRANK S J; GOODMAN N; LEE L Q; MARGULIES B I; WEBER F D

Number of Countries: 013 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 1016971	A2	20000705	EP 88311139	A	19881124	200128	B
			EP 2000200994	A	19881124		

Priority Applications (No Type Date): US 87136930 A 19871222

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 1016971	A2	E	51	G06F-012/08	Div ex application EP 88311139

Div ex patent EP 322117
Designated States (Regional): AT BE CH DE ES FR GB GR IT LI LU NL SE

Abstract (Basic): EP 1016971 A2

NOVELTY - Each processing cell (12) including a central processing unit (CPU) connected to an associated memory adapted to store data items. Each CPU generates an access request for **accessing** a **data** item, which includes a data item identifier for all of the data items in a system address space which is a unitary system address space. The type of request could be a retrieval or storage request type.

DETAILED DESCRIPTION - The system address space defines a number of pages at least one of which has a number of sub-pages, which contains at least one data item. Each sub-page maintained by a memory is associated with one of a number access states. The access states are an exclusive **owner access** state, invalid **access** state, non-exclusive access state and an atomic **owner access** state. The memory management system receives selected ones of the access requests, for **using** the **data** item identifier in each respective access request received to identify one of the processing cells whose memory has stored one of the sub-pages containing the data item identified by the data item identifier. A copy of the sub-page is transferred to the memory of the processing cell whose central processor unit generated the access request for storage in the memory.

INDEPENDENT CLAIMS are also included for the following: A digital data processing apparatus; and A method of operating a digital data processing apparatus.

USE - For **multiprocessing** systems with distributed hierarchical memory architectures.

ADVANTAGE - Provides an improved mutinous system with improved data coherency, as well as reduced latency, bus contention and also with unlimited scalability.

DESCRIPTION OF DRAWING(S) - The figure shows the structure of a **multiprocessing** system.

Processing cells (12)

pp; 51 DwgNo 1/9

Title Terms: MEMORY; MANAGEMENT; DIGITAL; DATA; PROCESS; CONTAIN; NUMBER; PROCESS; CELL; DATA; ITEM; IDENTIFY; DATA; ITEM; SYSTEM; ADDRESS; SPACE; UNIT; SYSTEM; ADDRESS; SPACE

Derwent Class: T01

International Patent Class (Main): G06F-012/08

International Patent Class (Additional): G06F-015/16

File Segment: EPI

Manual Codes (EPI/S-X): T01-H03B; T01-H05B1; T01-H07C7; T01-H08

26/9/11 (Item 11 from file: 350)

DIALOG(R)File 350:Derwent WPIX

013638007 **Image available**

WPI Acc No: 2001-122215/200113

XRPX Acc No: N01-089690

Direct data transfer providing method for multiprocessing **, involves transferring requested data to one local memory from the other, independent of data segment transfer to main memory and directory storage**

Patent Assignee: UNISYS CORP (BURS)

Inventor: BAUMAN M A; GILBERTSON R L; HAUPT M L

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week

US 6167489 A 20001226 US 98218811 A 19981222 200113 B

Priority Applications (No Type Date): US 98218811 A 19981222
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
US 6167489 A 25 G06F-013/00

Abstract (Basic): US 6167489 A
NOVELTY - The requested data segment is transferred to the main memory and the directory storage, to revise the ownership status to reflect a change of ownership from the primary device to the secondary device. The requested data segment is transferred to the secondary local memory from primary memory based on data transfer request, independent of the transfer of segment to the main memory and directory storage.
DETAILED DESCRIPTION - The method involves requesting data transfer of the requested data segment in the primary local memory to the secondary local memory of the secondary device. The requested data segment is transferred from the primary local memory to the secondary local memory of the secondary device in response to the request. The ownership status of the requested data segment is determined by **accessing** one or more **ownership** status bits corresponding to the requested data segment from the directory storage. A direct transfer of the requested data segment from the primary device to the secondary device is allowed when the ownership bits establish the primary device as the current owner of the requested data segment. An INDEPENDENT CLAIM is also included for system bypass supervisory memory intervention for data transfer between devices having associated local memories.
USE - In cache memory schemes for **use** in **multiprocessing** of **data** processing systems, shared memory systems, large scale data processing system.
ADVANTAGE - Allows requested data segments stored in primary device to bypass the main memory and travel directly from its current local memory to requesting device. Provides for the independent return of the data segment from its current local memory to the main supervisory memory to maintain memory coherency.
DESCRIPTION OF DRAWING(S) - The figure shows the flow diagram of bypass buffering method.
pp; 25 DwgNo 9/9
Title Terms: DIRECT; DATA; TRANSFER; METHOD; **MULTIPROCESSOR** ; TRANSFER; REQUEST; DATA; ONE; LOCAL; MEMORY; INDEPENDENT; DATA; SEGMENT; TRANSFER; MAIN; MEMORY; DIRECTORY; STORAGE
Derwent Class: T01
International Patent Class (Main): G06F-013/00
File Segment: EPI
Manual Codes (EPI/S-X): T01-H03A
? t26/9/13,15-17

26/9/13 (Item 13 from file: 350)
DIALOG(R)File 350:Derwent WPIX

012520935 **Image available**
WPI Acc No: 1999-327041/199927
XRPX Acc No: N99-245304
Multiprocessor **system with main memory having data ownership determination facility.**
Patent Assignee: UNISYS CORP (BURS)
Inventor: BAUMAN M A; MORRISSEY D E; NILSON A J
Number of Countries: 083 Number of Patents: 008

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9923566	A1	19990514	WO 98US23509	A	19981104	199927	B
AU 9913811	A	19990524	AU 9913811	A	19981104	199940	
US 6052760	A	20000418	US 97964626	A	19971105	200026	
EP 1027655	A1	20000816	EP 98957585	A	19981104	200040	
			WO 98US23509	A	19981104		
BR 9814846	A	20001003	BR 9814846	A	19981104	200053	
			WO 98US23509	A	19981104		
JP 2001522091	W	20011113	WO 98US23509	A	19981104	200204	
			JP 2000519359	A	19981104		
EP 1027655	B1	20040310	EP 98957585	A	19981104	200418	
			WO 98US23509	A	19981104		
DE 69822320	E	20040415	DE 622320	A	19981104	200426	
			EP 98957585	A	19981104		
			WO 98US23509	A	19981104		

Priority Applications (No Type Date): US 97964626 A 19971105

Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9923566	A1	E	33	G06F-012/08	
Designated States (National): AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GE GH GM HR HU ID IL IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG UZ VN YU ZW					
Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW NL OA PT SD SE SZ UG ZW					
AU 9913811	A			G06F-012/08	Based on patent WO 9923566
US 6052760	A			G06F-013/00	
EP 1027655	A1	E		G06F-012/08	Based on patent WO 9923566
Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE					
BR 9814846	A			G06F-012/08	Based on patent WO 9923566
JP 2001522091	W		37	G06F-015/177	Based on patent WO 9923566
EP 1027655	B1	E		G06F-012/08	Based on patent WO 9923566
Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE					
DE 69822320	E			G06F-012/08	Based on patent EP 1027655 Based on patent WO 9923566

Abstract (Basic): WO 9923566 A1

NOVELTY - In the system when data is requested by a requesting agent the main memory checks the history information associated with the requested data unit before forwarding the requested data unit.

DETAILED DESCRIPTION - The **multiprocessor** system (100) has a memory hierarchy which provides data **ownership** based upon previous **access** patterns of a **data** unit (230). The memory hierarchy comprises a main memory (210) having a number data units, a number of caches (510) that operate independently of each other, and at least one coherent domain (530) interfaced to each cache. A requesting agent (510,540,550) is allowed by the system to request a data unit without specifying the type of ownership, where ownership may be exclusive or shared. The directory (610), which is used by the main memory for maintaining information about each data unit, includes history information that defines the previous access pattern of the requested data unit. The main memory checks, prior to forwarding the requested data unit to the requesting agent, the history information to determine what type of ownership to associate with the requested data unit. The requested data unit is then delivered to the requesting agent with ownership rights specified by the history information.

USE - The system provides data **ownership** based upon previous

access patterns of the **data** unit.

ADVANTAGE - The method of restricting **access** to **data** units based on history information that defines previous **access** patterns of requested **data** unit does not prevent efficient performance of software operations.

DESCRIPTION OF DRAWING(S) - The figure illustrates the embodiment of the directory based memory and cache system.

Main memory (210)
Data unit (230)
Cache memory (510)
Directory (610)
pp; 33 DwgNo 6/10

Title Terms: **MULTIPROCESSOR** ; SYSTEM; MAIN; MEMORY; DATA; DETERMINE; FACILITY
Derwent Class: T01
International Patent Class (Main): G06F-012/08; G06F-013/00; G06F-015/177
File Segment: EPI
Manual Codes (EPI/S-X): T01-H03A; T01-H03D; T01-M02

26/9/15 (Item 15 from file: 350)
DIALOG(R)File 350:Derwent WPIX

011302655 **Image available**
WPI Acc No: 1997-280560/199725
XRPX Acc No: N97-232502

Fault tolerant dual fetch and store method for multiprocessor **system - involves protecting data stored in each of memory portions by interlocking data protection mechanism so that only one of requestors can** access data **previously stored at any one time**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: KRYGOWSKI M A; SUTTON A J
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 5630045	A	19970513	US 94349768	A	19941206	199725	B

Priority Applications (No Type Date): US 94349768 A 19941206
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 5630045	A		13	G06F-011/34	

Abstract (Basic): US 5630045 A

The method involves performing an initialization step during which separate portions in the memory unit are set aside. Each portion is designated to each of the memory controllers for its use. Physical addresses of each the memory portion are storing in a logic array in the memory unit so that the memory portions can be **accessed** at any time. **Data** stored in each of the memory portions are protected by an interlocking data protection mechanism so that only one of the requestors can **access** the **data** previously stored at any one time. The interlocking data protection mechanism provides an exclusivity **ownership access** status for the unit of data so that it can only be accessed and used only by one requestor at any one time. Each of the memory controller process is allowed a fetch or a store request issued by the requestors concurrently, the memory controllers **access data** stored in their respective designated memory portion.

ADVANTAGE - Allows parallel stores in fault tolerant system. Improve performance and integrity fault tolerant system.

Dwg.5b/7
Title Terms: FAULT; TOLERATE; DUAL; FETCH; STORAGE; METHOD; **MULTIPROCESSOR** ; SYSTEM; PROTECT; DATA; STORAGE; MEMORY; PORTION; INTERLOCKING; DATA; PROTECT; MECHANISM; SO; ONE; CAN; ACCESS; DATA; STORAGE; ONE; TIME
Derwent Class: T01
International Patent Class (Main): G06F-011/34
File Segment: EPI
Manual Codes (EPI/S-X): T01-G05C1

26/9/16 (Item 16 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010876112 **Image available**
WPI Acc No: 1996-373063/199638
XRPX Acc No: N96-313887
Enforcing exclusive access **to shared** resources **in** multi - tasking **computer - provides fast crash safe** semaphore **mechanism to enforce** mutually exclusive access **to shared** resources **, implemented at user level by providing common library code for all processes controlling acquisition and release of** semaphores
Patent Assignee: AT & T IPM CORP (AMTT); AT & T CORP (AMTT); AMERICAN TELEPHONE & TELEGRAPH CO (AMTT); LUCENT TECHNOLOGIES INC (LUCE)
Inventor: BOHANNON P L; GAVA J; LIEUWEN D F; SUDARSHAN S; SUDARSHAR S
Number of Countries: 007 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 727742	A2	19960821	EP 96300823	A	19960207	199638	B
CA 2165493	A	19960818	CA 2165493	A	19951218	199650	
JP 8286935	A	19961101	JP 9627619	A	19960215	199703	
US 5623670	A	19970422	US 95390179	A	19950217	199722	

Priority Applications (No Type Date): US 95390179 A 19950217
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 727742	A2	E	23	G06F-009/46	
Designated States (Regional): DE FR GB SE					
JP 8286935	A		27	G06F-009/46	
US 5623670	A		22	G06F-013/14	
CA 2165493	A			G06F-015/78	

Abstract (Basic): EP 727742 A

The computer comprises processors (15), and several shared resources, including a memory (30) accessible by each of the processors. Several **semaphore** records are stored in the memory, each of which is associated with one of the shared resources. The records include availability indicia indicating whether the record is available, owner identification registering ownership of the record, and cleanup indicia indicating whether the record is being evaluated by a cleaning mechanism.

Several processes are being executed on the processors requesting and acquiring exclusive **access** to the shared resources by requesting and **acquiring ownership** of the **semaphore** records. Several **semaphore** access records are stored in the memory, each one associated with one of the processes for identifying a **semaphore** record, which the process associated with the record wants to **acquire ownership** of. The clean-up mechanism determines whether a record is owned by a process which has crashed by using the information in the records and the access records.

ADVANTAGE - Provides for **semaphores** implemented at user level while allowing system to determine when failed process holds **semaphore** regardless of when process may have failed.

Dwg.1/8

Abstract (Equivalent): US 5623670 A

A computer system comprising:

at least one processing unit;

a plurality of shared resources, including a memory unit, accessible by said at least one processing unit;

a plurality of **semaphore** records stored in said memory unit, each of said **semaphore** records associated with one of said plurality of shared resources, wherein each of said **semaphore** records comprises:

a) availability indicia means for indicating whether the **semaphore** record is available,

b) owner identification means for registering ownership of the **semaphore** record, and

c) cleanup indicia for indicating whether the 1semaphore record is being evaluated by a cleanup routine;

a **plurality** of **processes** executing on said plurality of processing units, said processes requesting and acquiring exclusive access to said plurality of shared **resources** by requesting and **acquiring ownership** of said **semaphore** records; and

a plurality of **semaphore** access records stored in said memory unit, each of said **semaphore** access records associated with one of said **plurality** of **processes** , for identifying a **semaphore** record which said process associated with said **semaphore** access record wants to **acquire ownership** of;

wherein said cleanup routine comprises means for determining whether a **semaphore** record is owned by a process that has crashed by using information in said **semaphore** records and said **semaphore** access records and means for resetting the availability indicia of a **semaphore** record when the **semaphore** record is owned by a process that has crashed.

Dwg.6/8

Title Terms: ENFORCE; EXCLUDE; ACCESS; SHARE; RESOURCE; MULTI; COMPUTER; FAST; CRASH; SAFE; **SEMAPHORE** ; MECHANISM; ENFORCE; MUTUAL; EXCLUDE; ACCESS; SHARE; RESOURCE; IMPLEMENT; USER; LEVEL; COMMON; LIBRARY; CODE; PROCESS; CONTROL; ACQUIRE; RELEASE

Derwent Class: T01

International Patent Class (Main): G06F-009/46; G06F-013/14; G06F-015/78

International Patent Class (Additional): G06F-015/16

File Segment: EPI

Manual Codes (EPI/S-X): T01-F02C; T01-H03D; T01-H08; T01-M02

26/9/17 (Item 17 from file: 350)

DIALOG(R)File 350:Derwent WPIX

010635543 **Image available**

WPI Acc No: 1996-132496/199614

XRPX Acc No: N96-111410

Multiprocessor **system - operating on red structure i.e. in display component which shows number to be displayed corresp. to cache under one process**

Patent Assignee: TOSHIBA KK (TOKE)

Number of Countries: 001 Number of Patents: 001

Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 8006805	A	19960112	JP 94132878	A	19940615	199614	B

Priority Applications (No Type Date): JP 94132878 A 19940615
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
JP 8006805 A 7 G06F-009/46

Abstract (Basic): JP 8006805 A

The system has an exclusion control range recognition unit which acknowledges the **data** shared and **access** in bind state process. The affiliation process recognises the execution of **data** shared and **access** and the number of **data** being shared and **accessed** is also acknowledged.

The number to be displayed on a red structure such as display component is arranged such that proper light corresponds to it. A lock operation is performed in the critical domain which **accesses** the shared **data owned** jointly between by systems.

ADVANTAGE - Excludes unnecessary lock control. Carries out effective use of processor by optimum lock control.

Dwg.1/5

Title Terms: **MULTIPROCESSOR** ; SYSTEM; OPERATE; RED; STRUCTURE; DISPLAY; COMPONENT; SHOW; NUMBER; DISPLAY; CORRESPOND; CACHE; ONE; PROCESS
Derwent Class: T01
International Patent Class (Main): G06F-009/46
International Patent Class (Additional): G06F-015/16
File Segment: EPI
Manual Codes (EPI/S-X): T01-H03D; T01-M02
? t26/9/19,29

26/9/19 (Item 19 from file: 350)
DIALOG(R)File 350:Derwent WPIX

008504986 **Image available**
WPI Acc No: 1991-009070/199102
XRPX Acc No: N91-007115

Dynamic resource pooling data processing system - management of computer system resources applied to multiple processes **concurrent execution**

Patent Assignee: IBM CORP (IBMC)
Inventor: DONG M A; TREIBER R K
Number of Countries: 004 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 405724	A	19910102	EP 90304688	A	19900430	199102	B
US 5093912	A	19920303	US 89371563	A	19890626	199212	
EP 405724	A3	19930224	EP 90304688	A	19900430	199348	

Priority Applications (No Type Date): US 89371563 A 19890626
Cited Patents: NoSR.Pub; 6.Jnl.Ref
Patent Details:
Patent No Kind Lan Pg Main IPC Filing Notes
EP 405724 A
Designated States (Regional): DE FR GB
US 5093912 A 25

Abstract (Basic): EP 405724 A

The data processing system in which resource allocation is achieved by one of the available processors from a master resource pool of default capacity from which **resources** are allocated for **use** and to which **used resources** are returned when they have been freed by their current user, the master resource pool can then be expanded and

additional resources made available for allocation by being added to the master resource pool, from an extension resource pool, created when needed, these additional resources being tagged as belonging to that specific extension resource pool and are returned to it, rather than to the master pool, once freed and that extension resource pool is logically deleted once it has had all its resources returned to it, effectively contracting the master resource pool.

A management technique to manage a pool of resources in a multiprogramming or **multiprocessing** environment includes the expansion and contraction of the resource poll without making the mainline **use** of the **resources** inefficient and without requiring the use of latches or other extrinsic serialisation mechanisms in the mainline; thus this technique can be used by high performance programs that must be able to operate efficiently and by programs that must be able to get and free resources when executing in disabled mode or without interruption.

USE/ADVANTAGE - **Multiprocessing** and multiprogramming environments; minimises use of serialisation. (29pp Dwg.No.1/2)

Abstract (Equivalent): US 5093912 A

The method involves expanding and contracting a resource pool dynamically, i.e. in response to demand for the resources. **Resources** are allocated for **use** from a single master pool (22). The overall pool of available resources is expanded by creating an extension pool and adding its resources to the master pool. The master pool can theoretically be expanded to the limits of the number of resources available in the system.

To contract the pool dynamically first one of the extension pools is selected to be freed (FREE...POOL 66). Those resources owned by the freed pool but still in the master pool are moved to the extension pool, preventing their reuse. **Resources** in **use** which are **owned** by the freed pool are labelled by changing their home address (52) so that they will be returned to the extension pool instead of to the master pool. Once all such resources have been returned to the freed extension pool it is deleted, thereby contracting the overall pool of resources.

USE - For **data** processing system.

Dwg.2/9

Title Terms: DYNAMIC; RESOURCE; DATA; PROCESS; SYSTEM; MANAGEMENT; COMPUTER ; SYSTEM; RESOURCE; APPLY; MULTIPLE; PROCESS; CONCURRENT; EXECUTE

Derwent Class: T01

International Patent Class (Additional): G06F-009/46; G06F-012/00

File Segment: EPI

Manual Codes (EPI/S-X): T01-F02

26/9/29 (Item 29 from file: 347)

DIALOG(R)File 347:JAPIO

03582357 **Image available**

ACQUISITION METHOD FOR **OWNERSHIP** OF **MULTIPROCESSOR** SYSTEM RESOURCES

PUB. NO.: 03-245257 [JP 3245257 A]

PUBLISHED: October 31, 1991 (19911031)

INVENTOR(s): KUDO MASASHI

APPLICANT(s): NEC CORP [000423] (A Japanese Company or Corporation), JP (Japan)

APPL. NO.: 02-043157 [JP 9043157]

FILED: February 23, 1990 (19900223)

INTL CLASS: [5] G06F-015/16

JAPIO CLASS: 45.4 (INFORMATION PROCESSING -- Computer Applications)

JOURNAL: Section: P, Section No. 1305, Vol. 16, No. 39, Pg. 57, January 30, 1992 (19920130)

ABSTRACT

PURPOSE: To keep the balance between a peripheral device and a processor by performing the input/output request processing with preference for an input/ output task having a running level higher than an application task.

CONSTITUTION: A main storage 3 includes a bus 4 which connects the processors 1 and 2 to the storage 3 and stores the exclusive resources 13 and 14, a resources acquring means 5 which **acquires** the **resources** 13 and 14 via the spin lock control, a resources release means 6 which releases the **acquires resources** , a task running level setting means 7 which **sets** the **task** running levels and allocates each task to its corresponding running level, and a resources processing running level transition means 8 which allocates the acquired tasks to each running level after the resources 13 and 14 are acquired via the spin lock control and then the task running levels are set for execution of the processing to those resources. Then a spin state is secured at a lower task running level even though a task 13 is set in a spin state by the processor 2 in a spin lock state. Therefore an input/output task having a higher running level than the task 13 receives an external interruption and processes an input/ output request with preference.
?

```
File   6:NTIS 1964-2004/Jul W3
         (c) 2004 NTIS, Intl Cpyrght All Rights Res
File   2:INSPEC 1969-2004/Jul W2
         (c) 2004 Institution of Electrical Engineers
File   8:Ei Compendex(R) 1970-2004/Jul W2
         (c) 2004 Elsevier Eng.  Info. Inc.
File 256:SoftBase:Reviews,Companies&Prods. 82-2004/Jun
         (c)2004 Info.Sources Inc
File  34:SciSearch(R) Cited Ref Sci 1990-2004/Jul W2
         (c) 2004 Inst for Sci Info
File  35:Dissertation Abs Online 1861-2004/May
         (c) 2004 ProQuest Info&Learning
File  65:Inside Conferences 1993-2004/Jul W3
         (c) 2004 BLDSC all rts. reserv.
File  94:JICST-EPlus 1985-2004/Jun W4
         (c)2004 Japan Science and Tech Corp(JST)
File  95:TEME-Technology & Management 1989-2004/Jun W1
         (c) 2004 FIZ TECHNIK
File  99:Wilson Appl. Sci & Tech Abs 1983-2004/Jun
         (c) 2004 The HW Wilson Co.
File 111:TGG Natl.Newspaper Index(SM) 1979-2004/Jul 20
         (c) 2004 The Gale Group
File 144:Pascal 1973-2004/Jul W2
         (c) 2004 INIST/CNRS
File 202:Info. Sci. & Tech. Abs. 1966-2004/Jul 12
         (c) 2004 EBSCO Publishing
File 233:Internet & Personal Comp. Abs. 1981-2003/Sep
         (c) 2003 EBSCO Pub.
File 266:FEDRIP 2004/May
         Comp & dist by NTIS, Intl Copyright All Rights Res
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
File 483:Newspaper Abs Daily 1986-2004/Jul 20
         (c) 2004 ProQuest Info&Learning
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
File 603:Newspaper Abstracts 1984-1988
         (c)2001 ProQuest Info&Learning

Set     Items   Description
S1    1094042   OWN OR OWNS OR OWNED OR OWNER? OR OWNING OR COOWN?
S2      75748   MULTIJOB? ? OR MULTIPROCESS?? ? OR MULTITHREAD? OR MULTITA-
              SK?
S3    4662718   THREAD?? ? OR PROCESSES OR JOB? ? OR TASK?? ? OR TASKING? ?
               OR PROGRAMMES OR PROGRAMS OR ENTITY? OR ENTITIES OR ROUTINE?
              ?
S4      79524   (MANY OR MULTIPLE OR MULTI OR SEVERAL OR NUMEROUS OR PLURA-
              LIT? OR MULTIPLE OR MULTITUD? OR PLURIF? OR BETWEEN)(1W)S3
S5      64922   (VARIOUS OR VARIETY OR GROUP???? ? OR CLUSTER? ? OR NUMBER
              OR SET OR SETS OR CHAIN? ? OR SERIES)(1W)S3
S6     114158   (TWO OR SECOND OR SECONDARY OR PAIR OR 2ND OR BOTH OR DUAL
              OR THREE OR TRIO OR THIRD OR 3RD)(1W)S3
S7     199663   LOCK OR LOCKS OR LOCKED OR LOCKING OR GHOSTLOCK? OR METALO-
              CK? OR SLEEPLOCK? OR SPINLOCK?
S8       8448   S1(3N)(INQUIR? OR QUERY? OR QUERIE? ? OR INTERROGAT? OR SU-
              BQUER? OR INQUIR? OR ENQUIR? OR ASK??? ? OR QUESTION??? ? OR -
              SEEK??? ?)
S9      29042   S1(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING
               OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCES-
              SING)
S10       116   S1(3N)(REUS???? ? OR REACQUISITION? OR REACCESS?)
```

S11 13616 MUTUAL?(1W)(EXCLUS? OR EXCLUD?) OR MUTEX? OR MUT()(EX OR EXES OR EXING OR EXED)
S12 1893 SEMAPHORE? OR EVENTSEMAPHORE?
S13 2045 CRITICAL()SECTION? ? OR CRITICAL()CODE? ?()SECTION? ?
S14 127 KERNELOBJECT? OR EVENTOBJECT? OR (KERNEL OR EVENT)()OBJECT? ?
S15 94388 RESOURCE? ?(1W)(MANAGE? OR MANAGING? OR DISPENS?)
S16 179018 SYNCHRONIS? OR SYNCHRONIZ?
S17 13032 S16(2N)(MECHANISM? OR OBJECT? ? OR STATE? ? OR PRIMITIVE? ? OR SERVICE? ? OR METHOD? ? OR STATEMENT? ?)
S18 10144 S16(2N)(OPERATION? ? OR FUNCTION? ? OR PROCESS?? ? OR DEVICE? ? OR UNIT OR UNITS)
S19 7755331 DATA OR RESOURCE OR RESOURCES
S20 939770 S19(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCESSING)
S21 433 S8:S10 AND (S2 OR MULTIPROCESS? OR S4:S6 OR MULTI()PROCESS?)
S22 16 S21 AND (S11:S15 OR S17:S18)
S23 21 S21 AND S11:S18
S24 68 S21 AND S20
S25 80 S22:S24
S26 21 S25/2000:2004
S27 59 S25 NOT S26
S28 45 RD (unique items)

28/7/13 (Item 2 from file: 2)
DIALOG(R)File 2:INSPEC

6140583 INSPEC Abstract Number: C1999-02-6150N-128
Title: Selective write-update: a method to relax execution constraints in a critical section
Author(s): Jae Bum Lee; Chu Shik Jhon
Author Affiliation: Dept. of Comput. Eng., Seoul Nat. Univ., South Korea
Journal: IEICE Transactions on Information and Systems vol.E81-D, no.11 p.1186-94
Publisher: Inst. Electron. Inf. & Commun. Eng,
Publication Date: Nov. 1998 Country of Publication: Japan
CODEN: ITISEF ISSN: 0916-8532
SICI: 0916-8532(199811)E81D:11L.1186:SWUM;1-Z
Material Identity Number: P713-1998-012
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: In a shared memory **multiprocessor**, shared **data** are usually **accessed** in a **critical section** that is protected by a lock variable. Therefore, the order of accesses by multiple processors to the shared data corresponds to the order of **acquiring** the **ownership** of the lock variable. The paper presents a selective write-update protocol, where data modified in a **critical section** are stored in a write cache and, at a **synchronization** point, they are transferred only to the processor that will execute the **critical section** following the current processor. By using QOLB **synchronization primitives**, the next processor can be determined at the execution time. We prove that the selective write-update protocol ensures data coherency of parallel programs that comply with release consistency, and evaluate the performance of the protocol by analytical modeling and program driven simulation. The simulation results show that our protocol can reduce the number of coherence misses in a **critical section** while avoiding the multicast of write-update requests on an interconnection network. In addition, we observe that

synchronization latency can be decreased by reducing both the execution time of a **critical section** and the number of write-update requests. From the simulation results, it is shown that our protocol provides better performance than a write-invalidate protocol and a write-update protocol as the number of processors increases. (16 Refs)
Subfile: C

28/7/15 (Item 4 from file: 2)
DIALOG(R)File 2:INSPEC

5119699 INSPEC Abstract Number: C9601-6150J-009
Title: Recoverable user-level mutual exclusion
Author(s): Bohannon, P.; Lieuwen, D.; Silberschatz, A.; Sudarshan, S.; Gava, J.
Author Affiliation: AT&T Bell Labs., Murray Hill, NJ, USA
Conference Title: Proceedings. Seventh IEEE Symposium on Parallel and Distributed Processing (Cat. No.95TB8131) p.293-301
Publisher: IEEE Comput. Soc. Press, Los Alamitos, CA, USA
Publication Date: 1995 Country of Publication: USA xvii+724 pp.
ISBN: 0 8186 7195 5
U.S. Copyright Clearance Center Code: 1063-6374/95/$04.00
Conference Title: Proceedings of Seventh IEEE Symposium on Parallel and Distributed Processing
Conference Sponsor: IEEE Comput Soc. Tech. Committee on Comput. Architecture; IEEE Comput. Soc. Tech. Committee on Distributed Process.; IEEE Comput. Soc. Dallas Chapter
Conference Date: 25-28 Oct. 1995 Conference Location: San Antonio, TX, USA
Language: English Document Type: Conference Paper (PA)
Treatment: Practical (P)
Abstract: **Mutual exclusion** primitives based on user-level atomic instructions (often called spin locks) have proven to be much more efficient than operating-system **semaphores** in situations where the contention on the **semaphore** is low. However, many of these spin lock schemes do not permit registration of ownership to be carried aut atomically with **acquisition** , potentially leaving the **ownership** undetermined if a process dies (or makes very slow progress) at a critical point in the registration code. We present an algorithm which can ensure the successful registration of ownership of a spin lock, regardless of where processes fail. Thus, our spin lock implementation is 'recoverable'. The determination of a spin lock's **ownership** can potentially be **used** to restore **resources** protected by the spin lock to consistency and then release the spin lock. Other processes using the lock can then continue to function normally, improving fault resiliency for the application. Our algorithm provides very fast lock acquisition when the acquisition is uncontested (comparable in speed to a simple test-and-set based spin lock), and we prove it works even on the weak memory consistency models implemented by many modern **multiprocessor** computer systems. (14 Refs)
Subfile: C

28/7/17 (Item 6 from file: 2)
DIALOG(R)File 2:INSPEC

03569538 INSPEC Abstract Number: B90017632, C90013899

Title: Shared memory in transputer systems
Author(s): Knowles, A.E.
Author Affiliation: Dept. of Comput. Sci., Manchester Univ., UK
Conference Title: IEE Colloquium on 'Transputer Applications' (Digest No.129) p.1/1-4
Publisher: IEE, London, UK
Publication Date: 1989 Country of Publication: UK 36 pp.
Conference Sponsor: IEE
Conference Date: 13 Nov. 1989 Conference Location: London, UK
Language: English Document Type: Conference Paper (PA)
Treatment: Applications (A)
Abstract: Although the use of shared memory goes against the ethos of the Occam model of computation, there are cases which justify its use in transputer systems. The author presents three such cases which arose within the ParSiFal Alvey project at Manchester University. Each has its **own** justification for the **use** of shared memory and uses it in a different way. In the first instance it was used as the vehicle for communication at an acceptable speed between a SUN system providing access to filestore and user console I/O. Secondly the bandwidth limitations of transputer links was overcome in the development of a card to allow real-time animation on a high resolution screen connected via links to a transputer network. Finally, advantage was taken of the clustering of transputers in the ParSiFal T-rack to provide alternative, higher speed, communication channels between transputers in a cluster by providing shared memory to which they all have access. Special care was exercised to allow the preservation of strict **synchronisation** between the communicating processes. (5 Refs)
Subfile: B C

28/7/20 (Item 9 from file: 2)
DIALOG(R)File 2:INSPEC

02777359 INSPEC Abstract Number: C87002339
Title: Fault-tolerant control of access **to shared** data **in cached-DASD storage systems**
Journal: IBM Technical Disclosure Bulletin vol.28, no.12 p.5457-9
Publication Date: May 1986 Country of Publication: USA
CODEN: IBMTAA ISSN: 0018-8689
Language: English Document Type: Journal Paper (JP)
Treatment: Practical (P)
Abstract: When two heat processors share data, a processor which is updating a portion of it must prevent the other processor from reading or updating that portion. A cached-DASD (direct-access storage device) system includes storage directors which place data into a subsystem memory or cache having a test-and-set register (TS), so that one processor only may 'own' TS and its data at once. The paper addresses the problem which arises when, due to a communication failure, the limiting caching or TS accessing director becomes nonoperational in that it prevents access to cache by the other director. The solution is to allow any operational director to 'steal' TS from the accessing director. This action requires a limited caching storage director to check before each directory **access** that it still ' **owns** ' TS since the other storage director may have 'stolen' it. (0 Refs)
Subfile: C
? t28/7/27

28/7/27 (Item 1 from file: 8)
DIALOG(R)File 8:Ei Compendex(R)

05206035 E.I. No: EIP99014537990

Title: Selective write-update: A method to relax execution constraints in a critical section

Author: Lee, Jae Bum; Jhon, Chu Shik

Corporate Source: Seoul Natl Univ, Seoul, South Korea

Source: IEICE Transactions on Information and Systems v E81-D n 11 Nov 1998. p 1186-1194

Publication Year: 1998

CODEN: ITISEF ISSN: 0916-8532

Language: English

Document Type: JA; (Journal Article) Treatment: A; (Applications); T; (Theoretical)

Journal Announcement: 9903W2

Abstract: In a shared-memory **multiprocessor** , shared **data** are usually **accessed** in a **critical section** that is protected by a lock variable. Therefore, the order of accesses by multiple processors to the shared data corresponds to the order of **acquiring** the **ownership** of the lock variable. This paper presents a selective write-update protocol, where data modified in a **critical section** are stored in a write cache and, at a **synchronization** point, they are transferred only to the processor that will execute the **critical section** following the current processor. By using QOLB **synchronization primitives** , the next processor can be determined at the execution time. We prove that the selective write-update protocol ensures data coherency of parallel programs that comply with release consistency, and evaluate the performance of the protocol by analytical modeling and program-driven simulation. The simulation results show that our protocol can reduce the number of coherence misses in a **critical section** while avoiding the multicast of write-update requests on an interconnection network. In addition, we observe that **synchronization** latency can be decreased by reducing both the execution time of a **critical section** and the number of write-update requests. From the simulation results, it is shown that our protocol provides better performance than a write-invalidate protocol and a write-update protocol as the number of processors increases. (Author abstract) 21 Refs.

?

```
File   9:Business & Industry(R) Jul/1994-2004/Jul 20
         (c) 2004  The Gale Group
File  16:Gale Group PROMT(R) 1990-2004/Jul 21
         (c) 2004 The Gale Group
File  47:Gale Group Magazine DB(TM) 1959-2004/Jul 21
         (c) 2004 The Gale group
File 148:Gale Group Trade & Industry DB 1976-2004/Jul 21
         (c)2004 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 275:Gale Group Computer DB(TM) 1983-2004/Jul 21
         (c) 2004 The Gale Group
File 570:Gale Group MARS(R) 1984-2004/Jul 21
         (c) 2004 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2004/Jul 21
         (c) 2004 The Gale Group
File 636:Gale Group Newsletter DB(TM) 1987-2004/Jul 21
         (c) 2004 The Gale Group
File 649:Gale Group Newswire ASAP(TM) 2004/Jul 19
         (c) 2004 The Gale Group

Set     Items   Description
S1    8598514   OWN OR OWNS OR OWNED OR OWNER? OR OWNING OR COOWN?
S2      82092   MULTIJOB? ? OR MULTIPROCESS?? ? OR MULTITHREAD? OR MULTITA-
             SK?
S3    7446632   THREAD?? ? OR PROCESSES OR JOB? ? OR TASK?? ? OR TASKING? ?
              OR PROGRAMMES OR PROGRAMS OR ENTITY? OR ENTITIES OR ROUTINE?
             ?
S4     171390   (MANY OR MULTIPLE OR MULTI OR SEVERAL OR NUMEROUS OR PLURA-
            LIT? OR MULTIPLE OR MULTITUD? OR PLURIF? OR BETWEEN)(1W)S3
S5     125174   (VARIOUS OR VARIETY OR GROUP???? ? OR CLUSTER? ? OR NUMBER
            OR SET OR SETS OR CHAIN? ? OR SERIES)(1W)S3
S6     139446   (TWO OR SECOND OR SECONDARY OR PAIR OR 2ND OR BOTH OR DUAL
            OR THREE OR TRIO OR THIRD OR 3RD)(1W)S3
S7     416024   LOCK OR LOCKS OR LOCKED OR LOCKING OR GHOSTLOCK? OR METALO-
            CK? OR SLEEPLOCK? OR SPINLOCK?
S8      64902   S1(3N)(INQUIR? OR QUERY? OR QUERIE? ? OR INTERROGAT? OR SU-
            BQUER? OR INQUIR? OR ENQUIR? OR ASK??? ? OR QUESTION??? ? OR -
            SEEK??? ?)
S9     350997   S1(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING
              OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCES-
            SING)
S10       959   S1(3N)(REUS???? ? OR REACQUISITION? OR REACCESS?)
S11     14424   MUTUAL?(1W)(EXCLUS? OR EXCLUD?) OR MUTEX? OR MUT()(EX OR E-
            XES OR EXING OR EXED)
S12      3564   SEMAPHORE? OR EVENTSEMAPHORE?
S13       839   CRITICAL()SECTION? ? OR CRITICAL()CODE? ?()SECTION? ?
S14       304   KERNELOBJECT? OR EVENTOBJECT? OR (KERNEL OR EVENT)()OBJECT?
             ?
S15    157578   RESOURCE? ?(1W)(MANAGE? OR MANAGING? OR DISPENS?)
S16    170694   SYNCHRONIS? OR SYNCHRONIZ?
S17      7160   S16(2N)(MECHANISM? OR OBJECT? ? OR STATE? ? OR PRIMITIVE? ?
              OR SERVICE? ? OR METHOD? ? OR STATEMENT? ?)
S18     11684   S16(2N)(OPERATION? ? OR FUNCTION? ? OR PROCESS?? ? OR DEVI-
            CE? ? OR UNIT OR UNITS)
S19  10910760   DATA OR RESOURCE OR RESOURCES
S20   1066006   S19(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USI-
            NG OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACC-
            ESSING)
S21      1766   S8:S10(S)(S2 OR MULTIPROCESS? OR S4:S6 OR MULTI()PROCESS?)
S22         8   S21(S)(S11:S15 OR S17:S18)
```

S23 13 S21(S)S11:S18
S24 13 S22:S23
S25 2 S24/2000:2004
S26 11 S24 NOT S25
S27 11 RD (unique items)

27/3,K/3 (Item 2 from file: 160)
DIALOG(R)File 160:Gale Group PROMT(R)

01485644
HITACHI INTRODUCE THE SMART APPROACH TO MULTI PROCESSOR COMMUNICATIONS.
NEWS RELEASE August 25, 1986 p. 11

... dual port RAM (SDPRAM), a highly intelligent DPRAM, which dramatically improves the communication efficiency of **multi - processor** systems by providing a communication path which is far superior in processing speed and functions...

... microprocessor bus interfaces which make it suitable for a particularly wide range of applications; additionally **semaphore** registers, which support **multi - processing** , control RAM access to provide an efficient means of controlling **access ownership** . Other features of the new smart DPRAM include four programmable FIFO status pins, which provide...

27/3,K/5 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01674922 SUPPLIER NUMBER: 15056167 (USE FORMAT 7 OR 9 FOR FULL TEXT)
A Windows NT C++ class for asynchronous I/O. (programming techniques) (includes related articles on asynchronous input/output details and use of Win32 application programming interface synchronization objects) (Tutorial) (Technical)
Tomlinson, Paula
Windows-DOS Developer's Journal, v5, n3, p25(14)
March, 1994
DOCUMENT TYPE: Technical ISSN: 1059-2407 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 5347 LINE COUNT: 00426

... when the object is signaled and are blocked when the object is not signaled.

A **mutex** (from the phrase " **mutual exclusion** ") is like a single-lane bridge that only one thread at a time can cross. If one thread successfully **acquires ownership** of a specific **mutex** , then no other threads can own it until the first thread releases it. **Mutexes** can be used to protect a global resource from being accessed by **multiple threads** at the same time. Note that a **mutex** (like a **semaphore** and an event) doesn't actually protect the resource directly; all threads wishing to access the resource must agree not to do so unless they have ownership of that particular **mutex** . **Mutexes** can be shared **between processes** .

Semaphores are like variable-lane bridges that allow a **number** of **threads** to cross at the same time. You specify how **many threads** can own the **semaphore** when you initialize it. Once the maximum **number** of **threads** have **acquired ownership** of a **semaphore** , other threads trying to own it are denied ownership until one of the owning threads releases it. **Semaphores** can be shared **between processes** .

Events are more like public broadcasts or triggers. Events are often

used when one thread...

27/3,K/7 (Item 4 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01629416 SUPPLIER NUMBER: 14622238 (USE FORMAT 7 OR 9 FOR FULL TEXT)
The fundamentals. (part 2) (Writing OpenVMS Privileged Code.) (Tutorial)
Goatley, Hunter; Heinrich, Edward A.
Digital Systems Journal, v15, n6, p38(4)
Nov-Dec, 1993
DOCUMENT TYPE: Tutorial ISSN: 1067-7224 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3391 LINE COUNT: 00272

... and one of the most commonly used, the I/O database.
Like a spinlock, a **mutex** is a flag that indicates whether or not a code thread is accessing the structure. The low word of the 31-bit **mutex** consists of the ownership count for a **mutex** ; the high word consists of a status word, indicating whether or not the **mutex** has been acquired for writing. In OpenVMS/VAX, the **mutex** count field is initialized to -1 so that a single TSTW instruction can be used to determine if the **mutex** is unowned, has exactly one **owner** , or is being **accessed** by **multiple** code **threads** . The global longword for the I/O database **mutex** has the symbolic name IOC$GL...

... **MUTEX** . Likewise, the logical name table **mutex** is LNM$AL...
... **MUTEX** , the paged dynamic memory **mutex** is EXE$GL

27/3,K/8 (Item 5 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01509494 SUPPLIER NUMBER: 12048390 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Abstract data types in C. (hidden data structures increase reusability, portability and readability) (Technical)
Ford, Dan
Computer Language, v9, n5, p59(7)
May, 1992
DOCUMENT TYPE: Technical ISSN: 0749-2839 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 2798 LINE COUNT: 00212

... common operations is not scattered or duplicated throughout the system. Another benefit is that, in **multitasking** systems, the abstract data type provides a convenient place to put **semaphores** that can allow it to perform its **own** serialized **access** to the data. That is, abstract data types can invisibly provide **mutual exclusion** if **multiple processes** need to access the same instance of the abstract data type.
Third, the approach encourages...

27/3,K/9 (Item 6 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01362721 SUPPLIER NUMBER: 08544538 (USE FORMAT 7 OR 9 FOR FULL TEXT)
OS/2 Version 2.0: exploiting the 32-bit architecture of 80386- and

80486-based systems.
Duncan, Ray
Microsoft Systems Journal, v5, n3, p1(14)
May, 1990
ISSN: 0889-9932 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 6793 LINE COUNT: 00528

... that uses semaphores and signals will need to be rethought and rewritten.

The 16-bit **semaphore** API has proven somewhat troublesome because it supports three subtly different types of **semaphores** -system **semaphores** , RAM **semaphores** , and Fast-safe RAM **semaphores** . Furthermore, the first two types can be used in two incompatible ways: for signaling an event, in which case the **semaphore** is either set or cleared, or for **mutual exclusion** , after which the **semaphore** is considered to be either owned or not owned. Overlap and ambiguity in the 16-bit AIPI functions adds to the confusion. For example, DosSemRequest **acquires ownership** of a **semaphore** , while DosSemSet sets a **semaphore** , but DosSemClear is used either to clear a **semaphore** used for signaling, or to release **ownership** of a **semaphore used** for **mutual exclusion** . The RAM **semaphores** can also be combined into lists, which require the use of still other special-purpose API functions, but only when the **semaphores** are used for signaling! Another chronic problem in the 16-bit **semaphore** API has been the status of system **semaphores** as a critical and often application limiting resource. System **semaphores** are valuable because they support counting or nested request calls, they are not swappable, they...

...and the operating system assists with clean up if a process dies owning a system **semaphore** . But these characteristics also ensure that relatively few system **semaphores** can be available in OS/2 1.x. System **semaphores** must be located below the 640kb boundary, and thus take away memory from the DOS...

...environment. In addition, OS/2 itself must use a large proportion of the available system **semaphores** . The 32-bit **semaphore** API solves these historical problems by tossing out the olds emaphore classes and functions completely. It defines three new types of **semaphores** - **mutual exclusion** (**mutex**) **semaphores** , event (signaling) **semaphores** , and multiple-wait (muxwait) **semaphores** and three new, disjoint sets of API functions to manipulate them. There is no longer any distinction between RAM **semaphores** and system **semaphores** , although it remains possible to create an anonymous **semaphore** that another process can't access; all **semaphores** are controlled by the system, are shareable, and can be cleaned up. There is also no longer any chance of **two processes** or threads coming into conflict in the way they access a **semaphore** ; if the wrong API call is used with a particular **semaphore** , an error code will be returned but no other damage will be done.

Signals as...

27/3,K/10 (Item 7 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)
(c) 2004 The Gale Group. All rts. reserv.

01208785 SUPPLIER NUMBER: 06077780 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Multiple tasks. (multitasking with OS-2)
Armbrust, Steven; Forgeron, Ted
PC Tech Journal, v5, n11, p90(9)
Nov, 1987
ISSN: 0738-0194 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT

WORD COUNT: 6693 LINE COUNT: 00517

... In the third case, the call returns immediately, indicating if the semaphore is currently owned.

Multiple threads can request ownership of the same **semaphore** . If the threads choose to wait (instead of returning immediately), the ones that do not immediately gain access to the **semaphore** continue to wait until the owning thread releases the **semaphore** . When this happens, the waiting threads are dispatched again, and the highest-priority waiting thread that has been waiting the longest becomes the new **owner** .

Using a **semaphore** to represent ownership of some resource enables threads to read or write sensitive data areas...

...Another thread reads the database and displays the aircraft positions on a screen. Without a **semaphore** guarding the database, the thread that writes the coordinates could write an airplane's x...

...could happen if the reading thread was interrupted before it read all three coordinates. Using **semaphores** , each thread requests ownership of the **semaphore** before reading or writing the database. Each would clear the **semaphore** after accessing the database to give other threads a chance in order to use or...

?

```
File 696:DIALOG Telecom. Newsletters 1995-2004/Jul 20
         (c) 2004 The Dialog Corp.
File  15:ABI/Inform(R) 1971-2004/Jul 20
         (c) 2004 ProQuest Info&Learning
File  98:General Sci Abs/Full-Text 1984-2004/Jun
         (c) 2004 The HW Wilson Co.
File 484:Periodical Abs Plustext 1986-2004/Jun W4
         (c) 2004 ProQuest
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File 613:PR Newswire 1999-2004/Jul 21
         (c) 2004 PR Newswire Association Inc
File 635:Business Dateline(R) 1985-2004/Jul 20
         (c) 2004 ProQuest Info&Learning
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 610:Business Wire 1999-2004/Jul 21
         (c) 2004 Business Wire.
File 369:New Scientist 1994-2004/Jul W2
         (c) 2004 Reed Business Information Ltd.
File 370:Science 1996-1999/Jul W3
         (c) 1999 AAAS
File  20:Dialog Global Reporter 1997-2004/Jul 21
         (c) 2004 The Dialog Corp.
File 624:McGraw-Hill Publications 1985-2004/Jul 20
         (c) 2004 McGraw-Hill Co. Inc
File 634:San Jose Mercury  Jun 1985-2004/Jul 20
         (c) 2004 San Jose Mercury News
File 647:CMP  Computer Fulltext 1988-2004/Jul W2
         (c) 2004 CMP Media, LLC
File 674:Computer News Fulltext 1989-2004/Jun W4
         (c) 2004 IDG Communications

Set     Items   Description
S1   10204579   OWN OR OWNS OR OWNED OR OWNER? OR OWNING OR COOWN?
S2      31106   MULTIJOB? ? OR MULTIPROCESS?? ? OR MULTITHREAD? OR MULTITA-
              SK?
S3    9318882   THREAD?? ? OR PROCESSES OR JOB? ? OR TASK?? ? OR TASKING? ?
               OR PROGRAMMES OR PROGRAMS OR ENTITY? OR ENTITIES OR ROUTINE?
              ?
S4     155171   (MANY OR MULTIPLE OR MULTI OR SEVERAL OR NUMEROUS OR PLURA-
              LIT? OR MULTIPLE OR MULTITUD? OR PLURIF? OR BETWEEN)(1W)S3
S5     142579   (VARIOUS OR VARIETY OR GROUP???? ? OR CLUSTER? ? OR NUMBER
              OR SET OR SETS OR CHAIN? ? OR SERIES)(1W)S3
S6     143727   (TWO OR SECOND OR SECONDARY OR PAIR OR 2ND OR BOTH OR DUAL
              OR THREE OR TRIO OR THIRD OR 3RD)(1W)S3
S7     524128   LOCK OR LOCKS OR LOCKED OR LOCKING OR GHOSTLOCK? OR METALO-
              CK? OR SLEEPLOCK? OR SPINLOCK?
S8     706552   S1(3N)(INQUIR? OR QUERY? OR QUERIE? ? OR INTERROGAT? OR SU-
              BQUER? OR INQUIR? OR ENQUIR? OR ASK??? ? OR QUESTION??? ? OR -
              SEEK??? ?)
S9     707791   S1(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USING
               OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACCES-
              SING)
S10       670   S1(3N)(REUS???? ? OR REACQUISITION? OR REACCESS?)
S11     18617   MUTUAL?(1W)(EXCLUS? OR EXCLUD?) OR MUTEX? OR MUT()(EX OR E-
              XES OR EXING OR EXED)
S12      2289   SEMAPHORE? OR EVENTSEMAPHORE?
S13       497   CRITICAL()SECTION? ? OR CRITICAL()CODE? ?()SECTION? ?
S14       137   KERNELOBJECT? OR EVENTOBJECT? OR (KERNEL OR EVENT)()OBJECT?
               ?
```

```
S15     159984   RESOURCE? ?(1W)(MANAGE? OR MANAGING? OR DISPENS?)
S16     109403   SYNCHRONIS? OR SYNCHRONIZ?
S17       4231   S16(2N)(MECHANISM? OR OBJECT? ? OR STATE? ? OR PRIMITIVE? ?
                 OR SERVICE? ? OR METHOD? ? OR STATEMENT? ?)
S18       6515   S16(2N)(OPERATION? ? OR FUNCTION? ? OR PROCESS?? ? OR DEVI-
                 CE? ? OR UNIT OR UNITS)
S19    8544615   DATA OR RESOURCE OR RESOURCES
S20     735077   S19(3N)(ACQUIR? OR REACQUIR? OR USE OR USES OR USED OR USI-
                 NG OR USAGE? ? OR ACQUISITION? OR ACCESS OR ACCESSE? ? OR ACC-
                 ESSING)
S21       1111   S8:S10(S)(S2 OR MULTIPROCESS? OR S4:S6 OR MULTI()PROCESS?)
S22          7   S21(S)(S11:S15 OR S17:S18)
S23         13   S21(S)S11:S18
S24         13   S22:S23
S25          6   S24/2000:2004
S26          7   S24 NOT S25
S27          7   RD (unique items)
```

Patent Assignment Abstract of Title

Total Assignments: 1

Application #: 09480390 **Filing Dt:** 01/11/2000 **Patent #:** NONE **Issue Dt:**

PCT #: NONE **Publication #:** NONE **Pub Dt:**

Inventor: Michael P. Wagner

Title: System, Device, and method for providing mutual exclusion for computer system resources

Assignment: 1

Reel/Frame:	Received:	Recorded:	Mailed:	Pages:
010599/0639	03/15/2000	02/14/2000	05/08/2000	4

Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).

Assignor: WAGNER, MICHAEL P. **Exec Dt:** 02/03/2000

Assignee: EMC CORPORATION
171 SOUTH STREET
HOPKINTON, MASSACHUSETTS 01748

Correspondent: BROMBERG & SUNSTEIN LLP
JEFFREY T. KLAYMAN
125 SUMMER STREET
BOSTON, MA 02110

Search Results as of: 7/20/2004 2:07:30 P.M.

If you have any comments or questions concerning the data displayed, contact OPR / Assignments at 703-308-9723
Web interface last modified: Oct. 5, 2002

h g h ff b